TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Veli Lumme

Intelligent Interpretation of Machine Condition Data

Julkaisu 1096 • Publication 1096

Tampere 2012

Tampereen teknillinen yliopisto. Julkaisu 1096
Tampere University of Technology. Publication 1096

Veli Lumme

Intelligent Interpretation of Machine Condition Data

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Konetalo Building, Auditorium K1702, at Tampere University of Technology, on the 10th of December 2012, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2012

ISBN 978-952-15-2963-4 (printed)
ISBN 978-952-15-2970-2 (PDF)
ISSN 1459-2045

ABSTRACT

This dissertation argues that classification is an effective tool in the prediction of machine condition. A system based on continuous learning can be developed to automate the laborious process of interpretation of symptoms derived from collected data into various normal and fault modes. In order to defend these arguments, the study seeks to explain, how prediction works and how the results can be evaluated.

The study explores the philosophy of condition monitoring in assuring the safe and uninterrupted operation of machines. Condition monitoring provides essential information for the maintenance and operability of process plants. Vibration monitoring is considered as one of the most important techniques to offer adequate and reliable information to maintain rotating machines in a condition, where they can perform their required functions without failure for a specified time period, when used under specified conditions.

In addition of detection and collection of data that indicate the state of a machine, condition monitoring includes the examination of symptoms and syndromes to determine the nature of faults or failures. High confidence level is required in both diagnostics and prognostics, because misinterpretation of condition related data may lead into severe economic consequences.

The diagnostics of machine condition is laborious and challenging. It requires a lot of analyst's effort and time to detect an anomaly and even more to identify a fault mode. This study presents methods to predict the current condition of machines using training data collected from various normal and fault modes on the same machine or substantially similar machines. Learning algorithms offer possibilities to increase the confidence level of prediction.

The study presents results on practical experiments to demonstrate the principles of continuous learning processes. The experiments rely on data collected from wind turbine gearboxes, which are extremely difficult to be diagnosed, because of the large amount of data and symptoms. The study proofs that significant improvements to current confidence level of prediction can be achieved by the use of learning systems.

PREFACE

The remote and automated diagnostics has fascinated me since I received my M.Sc. degree from the Helsinki University of Technology in 1980. At that time, most of the vibration analysis was performed by the usage of sweeping filter spectrum analysers, which was a major improvement to the pure time domain analysis. Fast Fourier Transform (FFT) was implemented in some analysers, but the instruments were not portable to be used in standard route collection. Tape recorders were often used to collect the vibration signals from the field and to play back the signals into the FFT analyser, but it was laborious. There appeared to be no effective means to diagnose vibration data from a remote location automatically.

The real revolution occurred, when portable vibration data collectors entered the industrial market in mid 80s. The first instruments, such as the Microlog from Palomar Technologies, offered very limited features and capabilities compared with the modern ones. Even as such, they really changed the world. Combined with data management software, the systems were used to conveniently and effectively collect, analyse and store vibration data. The collected data was easily available for further analysis and trending. Data related to fault modes was stored for reference purposes in a digital format, which ensured the maintenance of data quality for long periods. It was easy to export data from a condition monitoring system and import it into a similar system, which allowed the data to be analysed and diagnosed at several locations at the same time. Later, when email messaging became common, the data could also be transferred without a major delay.

Not much later, the first efforts to automate the diagnostic process appeared. These were ambitiously referred to as expert systems or knowledge based systems. The idea was to interpret the analysts' knowledge and expertise to common rules. In some cases, the analysts were interviewed to learn, how fault modes appear in the vibration data. This knowledge was then programmed into a condition monitoring software as a combination of rules. The creation and maintenance of the rule database was too laborious compared with the benefits and such systems really never penetrated the market.

Another tempting approach was also considered. While the condition monitoring knowledge and expertise was often concentrated among a few analysts, an idea came up to collect them to locations, where the data would be made available for further analysis. Such an approach was thought to be interesting for organizations with operations at distant, separate locations. I was personally involved in early 1990's in the development of a remote diagnostic centre for Gazprom, which operates more than 500 gas compressors along the gas pipelines. The gas compressor stations had only a limited capability to process and diagnose the collected data. It turned out that the amount of data to be transferred to a diagnostic centre would have been extremely challenging and even data communication via satellites were considered, but a satisfactory solution was not found at that time.

I was first introduced to neural networks in 1994 by Professor Pasi Koikkalainen, who suggested that a learning system could perhaps be used to predict the condition of a machine. With his valuable knowledge and assistance, we built a small experimental system to analyze data collected from a lifting crane in a smelting plant. The monitoring of a crane is extremely difficult, because the crane has many various operating states. The hook of the crane can be lifted or lowered, during which the travel speed is first accelerated, then constant and finally decelerated. The hook can be empty or it can be lifting an empty or full melting pot. Other factors affecting the vibration state include the transverse motion and crane operator’s measures. The experiment proved, however, that we could actually predict the current operational state by using the vibration descriptors as an input. This observation convinced us to file in 1996 the first patent application, which later turned out successful. I wish to thank both prof. Koikkalainen and B.Sc. Pertti Leinonen of Rautaruukki for their unprejudiced and encouraging attitude.

This invention lured my imagination, but my other occupations took me away from active development. The idea about the usage of learning systems remained, however, in my thoughts. While the invention seemed to offer a solution for monitoring of a single machine, it later became necessary to collect data from several machines experiencing various fault modes and combine this data into a common database. A European patent application was filed in 2000 for this invention and again successfully. A project was started to establish a remote diagnostic centre, where data from various machines at several locations was collected. The prediction was based on the combined training data. It was found that the system could predict a fault mode, when training data from a similar fault mode at another distant location was available. Unfortunately the funding of the project failed and despite the promising results, the project had to be terminated. I wish to thank my colleague and good friend B.Sc. Juha Seppä for his persistence and continued confidence in the development of this invention.

The idea of a remote diagnostic centre was not new at that time and several centres had already been established. The motive was to concentrate the knowledge of a few analysts in one location. It soon became clear that automated diagnosis was necessary, especially for online data at short measurement intervals. It was considered valuable to have an automated system to at least detect anomalies in the data flow, which would significantly reduce the work load.

As a peculiarity, the Finnish Maintenance Society constructed a website, where the analysts could share their data and diagnoses. While this was not really a diagnostic centre, it offered an open discussion forum. The website allowed data to be uploaded in various formats. The data could also be downloaded from the website, if the user happened to have the same condition monitoring software package.

The condition monitoring experts started talking about intelligent and expert systems. Smart sensors and e-maintenance were also discussed. However, there appeared to be no definition for smartness or intelligence in the context of condition monitoring. In my mind intelligence was to be understood as an ability to learn from previous occurrences of fault modes. Professor Seppo Virtanen encouraged and motivated me to start post-graduate studies in Tampere University of Technology, which offered me a new possibility to investigate this field again. I owe special thanks to him for his inspiration, encouragement and support.

The confidence of prediction was not yet tested with large data volumes. The fact that historical data could often be used to train a classifier to predict a normal and fault mode was not enough to prove that this would be true for all cases. The earlier studies relied on the descriptors derived from vibration signatures, which appeared to change significantly depending on the current process state. Also, the memorization operations needed improvement. Moventas offered a project, where large amounts of data collected from substantially similar machines were available. Most of the experiments for this study are based on this data. I am therefore very thankful to M.Sc. Jukka Elfström and M.Sc. Markus Pylvänen for their effort and devotion.

During the thirty years history in vibration monitoring, I have trained hundreds of analysts to excel in solving various problems in machine diagnostics. It is evident that an analyst has to understand the vibration basics, but it is equally important to learn new things about machines, symptoms and fault modes. Theory and practice have been made much easier to comprehend by the usage of interactive learning tools with simulators, case studies and virtual test rigs. Training with these tools is so much easier than before. I wish to thank B.Sc. Jason Tranter of Mobius Institute for providing these excellent methods to improve my training possibilities and my own understanding of vibration phenomena.

Training and learning are the topics of this thesis. They are the keys to success both in human and artificial intelligence. The trainer might be experienced and skilled and the training material could be excellent, but the adoption of knowledge should be verified. A certification program has been introduced to verify that vibration analysts, who are eligible for the examination and have successfully passed an examination, receive an internationally recognized proof of their professional competence. The examinations typically cover matters and questions that have been presented during the training course. At the time of writing this preface, more than 100 analysts in Finland have been certified. Again, I wish to thank Jason for making it possible for me to offer the Finnish analysts the opportunity to demonstrate their knowledge and expertise in machine diagnostics.

Prediction based on artificial intelligence is also based on learning. The confidence level of prediction can therefore be verified by comparing the predicted to the factual interpretation. Markus Pylvänen made a tremendous work in assisting me to perform this verification, for which I am extremely thankful.

My sincere thanks go to the pre-examiners of my thesis, Professor Heikki Koivo (Aalto University) and Professor Alasdair MacLeod (University of Highlands and Islands). I appreciate highly their remarkable comments that have benefited this study. I wish to extend my gratitude to Professor Jukka Heikkonen for his valuable and practical comments on the definitions related to neural networks and their application.

This work would not have been possible without the support of my close relatives and friends. In addition to sacrificing their time with me, they have asked me to explain my work in common language. When you are required to describe a problem and a solution understandably, you have to familiarize yourself even more thoroughly. I have received many new ideas especially concerning the future developments from my brother Martti Lumme, to whom I am utmost grateful. I wish to thank my children Marja and Ilkka and grandchildren Marjo and Elliot for bringing so much joy to my life. Finally, I am most grateful to my wife, Paula, for giving so much continuous love, attention and care to me, while I have been occupied with this work.

CONTENTS

LIST OF SYMBOLS

n	rotation speed
E	band energy
A	amplitude
A_v	amplitude failure limit
A_1	current amplitude
AI	anomaly index
x_{ij}	descriptor
z_{ij}	symptom
$\dot{x}_j$	mean of descriptor values
x_j	descriptor value
$\overline{x}$	mean data value in the population
s_i	standard deviation
y	membership
c	membership constant
s	Euclidean distance of the current sample from the class centre
s_{max}	Euclidean distance of the furthest training sample from the class centre
k	skew factor
l	layer index
L	time to failure
S	time to shutdown

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
CM	Condition Monitoring
CMaS	Moventas Condition Management System
FFT	Fast Fourier Transform
HP	High Pass
ISO	International Standardization Organization
LP	Low Pass
MDL	Minimum Description Length
NAN	Not A Number
RMS	Root Mean Square
SOM	Self-Organizing Map
TS-SOM	Tree-Structured Self-Organizing Map

1. INTRODUCTION

Monitoring of machine condition with various techniques produces significant amounts of data and information. The data as such does not reveal the current condition of a machine, but needs to be processed. Data processing typically includes extraction of descriptors and symptoms that are known or thought to be related to various failure mechanisms in a machine. For a particular fault some symptoms should be present, some might and others should not appear in the data. A single symptom cannot be used to evaluate machine condition. The combination of symptoms is known as a syndrome. When symptoms are effectively chosen to present failure progression, a syndrome presents a description of a machine condition. However, a syndrome can change, if the operating conditions are changed without fault progression. This brings an additional challenge to the data interpretation. It takes a lot of condition monitoring analyst’s time and effort to evaluate, if a machine is in a good or abnormal condition.

Even if machines are individual, the fault progression typically follows the same pattern. If you have seen the fault pattern once, you should be able diagnose the same fault persistently in the same or substantially similar machine. Machine maintenance is considered to have failed, if the same fault occurs repeatedly. Therefore, it should not be likely to see the same fault mode to progress to a failure. On the other hand, the same fault is more likely to happen in similar machines. Because of the diversity between machines, the same diagnostic process and evaluation may need to be repeated and the same effort taken to reach the correct data interpretation. Use of artificial intelligence might bring a new tool to effectively and reliably diagnose faults in a large population of similar machines in different environments.

This thesis relies on several inventions patented by the writer. The patent text discloses the general idea behind the invention, but often lacks the practical application. This thesis gives explanations proven by practical studies on, how to implement a continuously learning machine monitoring system that detects autonomously anomalies and faults.

The thesis is organized as follows. Chapter 2 introduces the principles of condition monitoring with a special attention to the new terminology presented in the ISO standards and the new methods developed for fault prediction using classification. In particular, the concept of intelligent diagnosis and prediction is pursued. The data and information types including the data processing are discussed in Chapter 3. The principles of classification are presented in Chapter 4. The main emphasis of this thesis is to show, how classification enables the continuous learning process in machine fault prediction. This is explained in a practical way in Chapter 5. This thesis includes a study of vibration data collected from various wind turbine gearboxes around the world. The interesting results of the study are presented in Chapter 6. The process states have a great impact on the condition monitoring descriptors and should therefore be considered in diagnosis of machine condition. Discussion on process states in prediction is presented in Chapter 7. This study revealed several potential future development possibilities that are discussed in Chapter 8. Finally the conclusions are presented Chapter 9.

The main contributions of this thesis are:

1. The introduction of a method for a continuous learning diagnostic system to automate the anomaly and fault detection processes.
2. The idea of generalization by the use of data from several environments and failure patters to build a common database for future predictions.
3. The study of large database to demonstrate the implementation of a continuous learning diagnostic system.
4. The presentation of tests to evaluate the confidence level of prediction of an automated diagnostic system.
5. The idea of discovering the relationship between the condition monitoring and process monitoring parameters.

2. CONDITION MONITORING

2.1. General

There are many different and even conflicting definitions for condition monitoring (CM) in various standards and textbooks. ISO 13372 defines condition monitoring accurately as the detection and collection of information and data that indicate the state of a machine. [ISO 13372, 2004]. CM can therefore be considered as an information provider for timely plant maintenance and operation. This information is often vital to the safe and economical operation of a single machine or the whole plant. It is a common opinion that CM aims to define the current condition of all monitored machines. While this is partly true, it lacks another main objective. CM should be used to predict the condition of all monitored machines at any selected time in future. Knowing the current condition only, might be too late to take necessary maintenance actions in time.

Availability is defined as the probability that a machine will, when used under specified conditions, operate satisfactorily and effectively [ISO 13372, 2004]. To achieve the requested availability the probability of failure should be maintained below the tolerable level. Therefore the shutdown should take place before any failure caused by any of the primary, secondary or even tertiary faults.

The machine state deteriorates, if faults or failures occur. Fault is a condition of an item that occurs when one of its components or assemblies degrades or exhibits abnormal behaviour, which may lead to the failure of the machine. Failure is a termination of the ability of an item to perform a required function [ISO 13372, 2004]. Condition monitoring is used to plan correct times and types of maintenance actions in order to restrict a fault progress to a failure. If an analyst is familiar with the fault progression, he or she can more reliably predict the severity of a fault and remaining safe lifetime.

Analysts involved with condition monitoring are required to make demanding decisions, which may have serious consequences. Both under- and over-diagnosis can be costly and risky. If a machine is run to breakdown, serious primary and secondary damages may occur with losses in production, revenues and personnel safety. If on the other hand the machine is stopped for maintenance prematurely, an unnecessary loss of production is caused. In general the condition monitoring analyst is required to evaluate, if the plant consisting of all the critical machines can be safely used until the next shutdown or if the shutdown should be advanced. In order to make this evaluation an analyst needs accurate and repeatable data. The general guidelines for the collection and analysis of condition monitoring data can be found in [ISO 17359, 2011] and [Nohynek, 2004].

Condition monitoring often relies on trending of descriptor values. Figure 1 presents the objective of CM in prediction of risk of failure at a given time in the future. Curve 1 shows the current probability of failure for the primary fault based on a descriptor derived from data. Curves 1a, 1b and 1c mark the alternative prediction hypotheses for the primary fault. Curve 2 marks the prediction of a secondary fault caused by the primary fault. The figure reveals that an accurate prediction of machine condition is a challenging task with many unknown variables. For instance, the evaluation of probability of failure is often based on a single descriptor. For some fault modes, such as mass imbalance, a linear dependency can be defined between the fault severity and the descriptor value. For many other fault modes, for instance bearing faults, such dependency may not exist. In fact, one should derive and trend several independent descriptors to cover all potential fault modes.

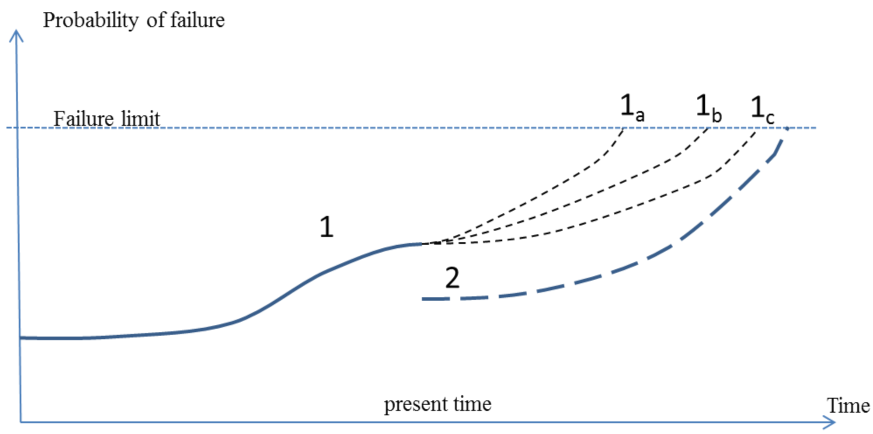

Figure 1: Trending of descriptors to predict the crossing of the failure limit. 1_a, 1_b and 1_c mark the alternative developments of the primary fault and number 2 the development of a secondary fault.

A second difficulty is in the setting of the failure limit. The crossing of a failure limit should alert the user on the significant possibility of a failure. In reality, it is seldom possible to set a definite failure limit, below which the probability of failure would be essentially lower than above it. Honestly, the actual time of failure cannot be predicted accurately. We should be contended to predict the crossing the failure limit.

Successful condition monitoring relies on the analysis of reliable and repeatable data, which is collected at various times (temporal data) and several locations (spatial data). Condition related data may be collected from various sources and in several quantities, such as vibration, temperature, oil analysis, acoustic emission, ultrasound, etc. This thesis deals with vibration data, which is commonly considered as the most informative for the purpose of prediction of machine health.

2.2. Temporal data

For diagnostic and prognostic purposes data is preferably received from various stages of the fault progression. A detailed investigation that is initiated by an anomaly or fault detection can be called a *case* study. One or more *events* are associated to a single case. An event refers to an observation based on a measurement or other means that can be assumed to be related to the same case. It is recommended that each event is diagnosed separately or at least at times, when the fault mode or severity has changed.

There are one or more *data records* collected essentially at the same time associated to every event. The data may consist of measurement results or other recordings. Typical data records may include for instance vibration spectra, time domains, trends, photographs, video clips, sketches or sound samples. [PSK 5970, 2011]

Vibration time or frequency domain data itself are also representations of short term temporal data. See Figure 2. They both indicate the periodicity of vibration phenomenon, which is typically observed in the evaluation of machine condition. Time and frequency domains reveal different types of information in the vibration signal. Generally, a frequency domain gives an illustrative representation of the characteristics of periodical signals. Time domain on the other hand shows more clearly the presence of non-harmonic and random phenomena in the vibration signal. For clarity, the frequency units in Figure 2 have been replaced with the multiples of a given base frequency. In many cases, it is desirable to analyze, how the various frequency components in a spectrum relate to the running speed. This information only is often adequate to coarsely diagnose the fault mode.

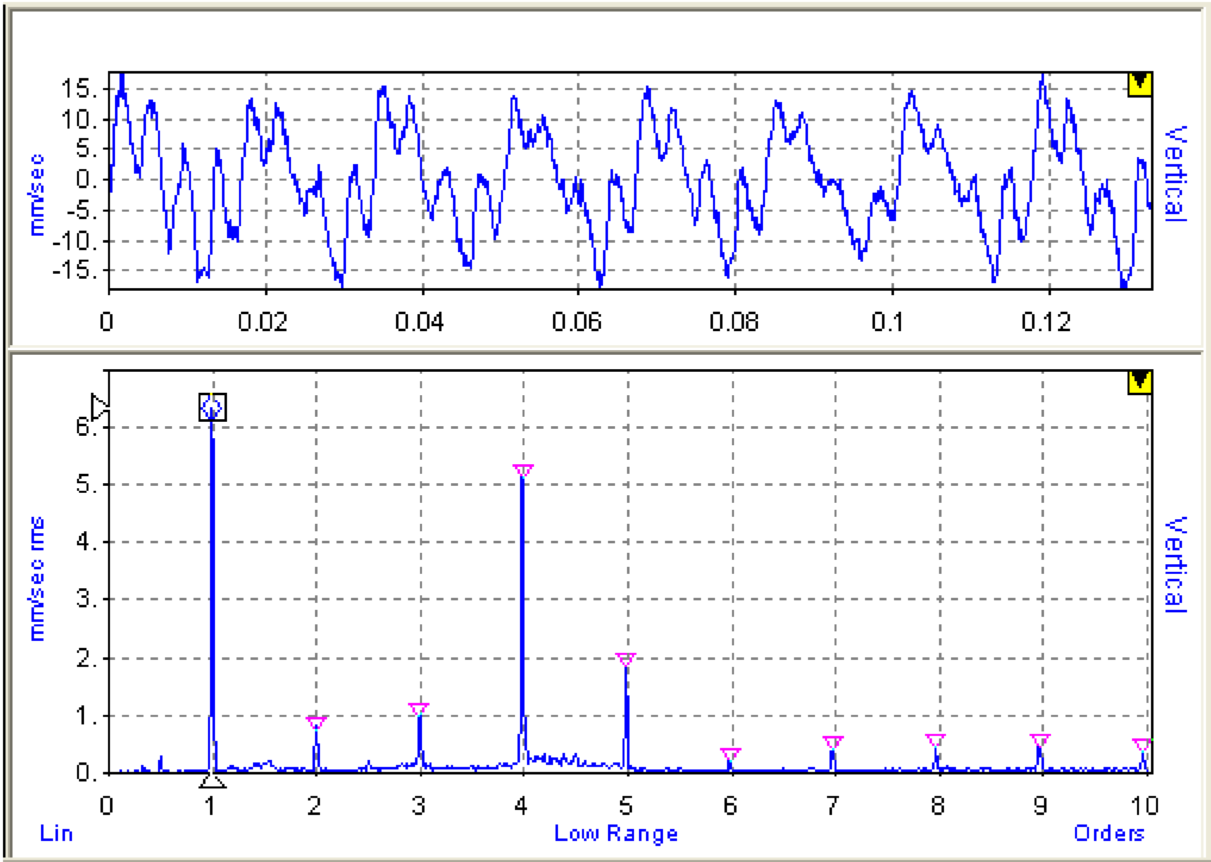

Figure 2: Vibration signal in time (above) and frequency (below) domains. The frequency peaks relate to characteristics of the cyclic motion of the machine.

For diagnostic and prognostic purposes, pre-failure data is needed as a baseline, which defines the normal behaviour of the machine. Failure data is used to determine the nature and severity of the fault. Post-failure data is useful in confirming that the problem has been solved and the machine was restored to its normal condition.

2.3. Spatial data

Spatial data is important in the determination, how the points of the machine move relatively. For this purpose, vibration values are measured at various locations and orientations. Especially, when phase readings are available, the relative motions between the various components can be visualized and easily understood. An example of such spatial data is given in Figure 3, which presents the amplitudes at various points shown as bars and relative phases as bubbles. This example shows that the vibration movements at the ends of the rotor are out-of-phase suggesting a potential couple imbalance.

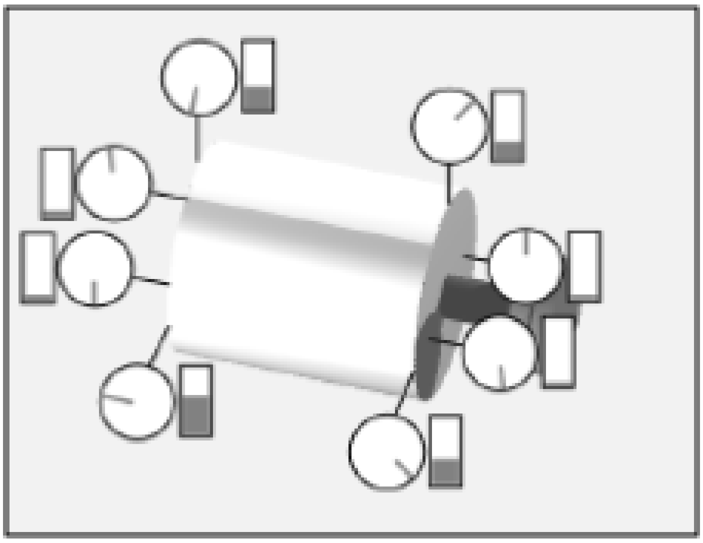

Figure 3: Vibration movements are out-of-phase at the rotor ends in radial direction

Usually data is taken under comparable conditions (load and speed), but this is not always possible, when the machine operates in a constantly changing environment. In such a case spatial data shall be collected regardless of the current environment and its impact to vibration behavior is taken into account separately. Spatial data may also include measurement of vibration values in similar or substantially similar machines nearby.

2.4. Descriptors, Symptoms and Syndromes

A *descriptor* is a data item derived from raw or processed parameters or an external observation. Descriptors are used to express symptoms and anomalies. The descriptors used for diagnostics are generally those obtained from the condition monitoring systems. However, operational parameters, such as other measurement, can also be considered as descriptors. A descriptor or group of descriptors forms a baseline data which provides a criterion of the normal behaviour of a machine under various process states. [ISO 13372, 2004].

In traditional condition monitoring systems, descriptors are used mainly for baseline and trending purposes. When used in classification, a descriptor (sometimes also referred to as a feature) should always have a numerical value. The selection of descriptors is very crucial and should be based on good machine knowledge and understanding of the various fault modes. When selected properly, any change in machine condition should be observed as a change in a respective descriptor value. An error in the selection of descriptors will lead into serious problems in the prediction of machine condition. Consequently, there should always be more than one descriptor in use to cover all potential fault modes.

The increased forces induced by various fault modes can be detected as changes in the vibration response measured typically at the bearings. The relationship between the forces and responses are often assumed to be linear, but that does not apply always. For instance in the case of a resonance, the change in vibration response can be multi-fold compared with the change in forces. However, adequate confidence level is usually achieved by using simple descriptors. Typically, groups of descriptors are obtained by post-processing various types of data. This process is often called a feature extraction. Some general rules for feature extraction apply especially when used in classification:

1. A descriptor should be sensitive to change, when machine state deteriorates.
2. Several descriptors are needed to cover all fault modes.
3. The descriptors should be derived either from the same data sample or samples obtained at precisely the same time.
4. The descriptors should be independent.
5. The descriptor definitions may need to be fuzzified.
6. Some descriptors should rather be used as explanatory variables.

While commonly used descriptors, such as overall vibration magnitude, might be extremely sensitive to changes in the severity of a fault mode, such as imbalance, it is important to acknowledge that the same descriptor might not react at all for another fault mode. Sometimes, the magnitude of a descriptor may even descent, when the fault progresses. In any case, it is crucial that at least one of the selected descriptors reacts significantly to the appearance of a fault.

In a condition monitoring system, data may be collected from various sources and locations at different times. For a human expert it might be quite easy to proportion the time lags between the descriptors, but for a classifier this is extremely difficult. If the descriptors should have been derived from data taken at various stages of a fault mode, the classifier would most probably give an ambiguous result. In a practical application the descriptors should rather be extracted from a same data source, such as vibration time domain or spectrum collected essentially at the same time.

Many of the descriptors used in a conventional condition monitoring systems are likely to be related. For many fault modes, vibration overall level is highly dependent of amplitude at rotating speed. If both the overall level and amplitude were used as classification descriptors, this would use unnecessarily the computation resources without any major advantages. Fault modes have different appearances and they may vary along with the fault progression. Some faults create vibration at forcing frequencies, while others result in broadband vibration energy. Some appear at low frequencies and others in high frequencies. Consequently, the amount and type descriptors should be adequately selected to cover the known fault modes.

The following lists some of the typical descriptors used:

1. amplitude at the rotating frequency (1 n)
2. amplitude at twice the rotating frequency
3. amount of harmonic vibration
4. amount of non-harmonic vibration
5. amount of sub-harmonic vibration
6. amount of sub-synchronous vibration
7. amount of low frequency vibration
8. amount of middle frequency vibration
9. amount of high frequency vibration
10. amplitude at any other forcing frequency and its harmonics, such as blade pass or gearmesh frequencies
11. amplitude at sidebands

The frequency band descriptors (such as items 7, 8 and 9 in the list above) are very sensitive to react to changes in the shape of the vibration spectrum. These descriptors are usually calculated as a root mean square (rms) value from a vibration spectrum using the following equation:

$$E = \sqrt{\sum_{i=f_l}^{f_h} A_i^2} \qquad (1)$$

where E is the vibration energy within the frequency band, i is the spectrum line index, A is the vibration amplitude at a given spectrum line, f_l is the lower frequency limit spectrum line and f_h is the higher frequency limit spectrum line.

The feature extraction described above can also be accomplished without frequency domain using a simple network, such as the one in Figure 4. The network consists of selectable low and high pass filters and outputs four frequency band descriptors derived from a single time signal.

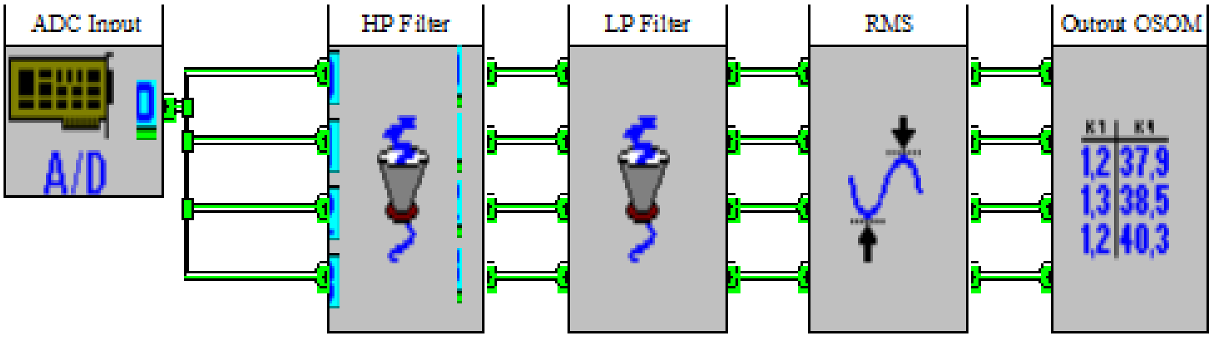

Figure 4: An example of a descriptor extraction network, where data is first acquired from an analog to digital converter module (ADC Input), then taken through high pass (HP Filter) and low pass (LP Filter) filter modules to a RMS Module that calculates the root mean square values for each frequency band. Finally the RMS values are output to a classification module (Output OSOM).

The frequency limits are often expressed relative to the rotating speed (n). Common guidelines for the frequency limits might be for instance 0 to 1.5 x n for low frequency vibration, 1.5 to 4 x n for middle frequency vibration and over 4 x n for high frequency vibration. This may appear as a perfect solution to cover all frequency ranges. However, there might be a frequency component close to some of the boundary frequencies. For instance, we might have a forcing frequency at approximately 1.5 x n. A small decrease in a forcing frequency would significantly increase the value of one descriptor and decrease the value of another descriptor. A small increase in forcing frequency would affect in the opposite direction. In classification, this insignificant frequency fluctuation would make a major difference. In order to avoid this, the frequency bands should be overlapping. As an example low frequency band for 0 to 1.7 x n, middle frequency band from 1.3 x n to 4.5 x n and high frequency band from 4 x n to maximum frequency. This method creates fuzzy borders between the adjacent descriptors, which again appear as small changes in the descriptor values caused by frequency fluctuation.

Figure 5 illustrates overlapping frequency bands. Each frequency band has a start and end frequency. Using the formula (1) the energy within the band can be calculated. The calculated band energy value is given in the vertical axis as amplitude values.

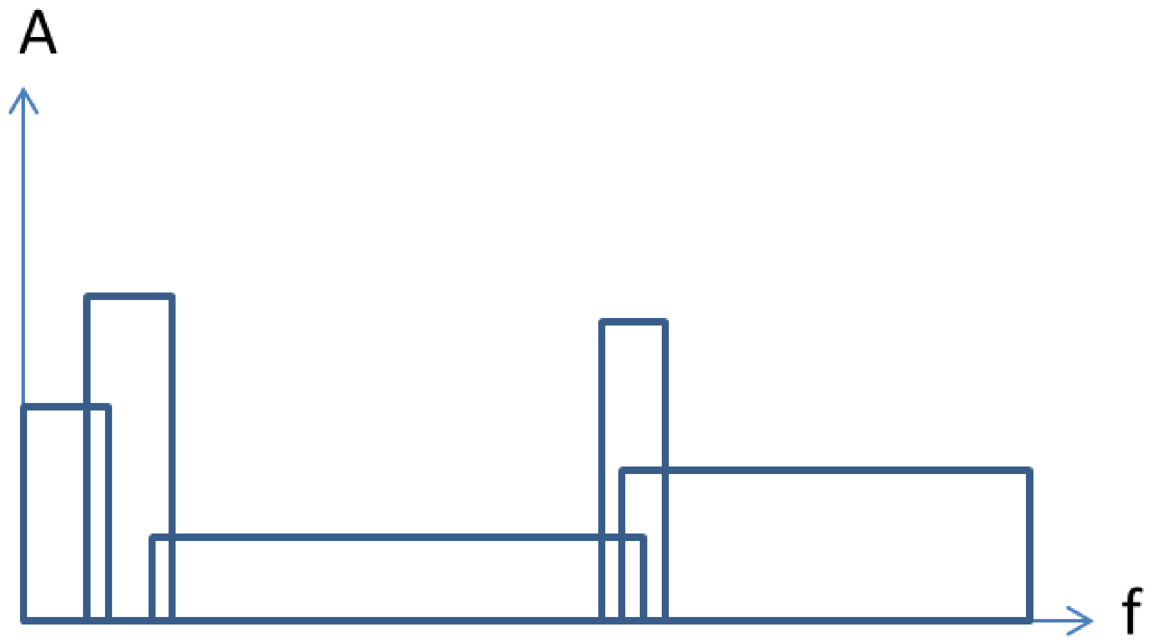

Figure 5: The start and end frequencies have been selected so that the adjacent frequency bands overlap. The heights of the frequency bands denote the calculated rms amplitudes within the band.

Some process parameters, such as rotation speed and load have a great impact on particular descriptors. It might be tempting to use these parameters as descriptors. However, they are usually not related to typical fault modes and should not be included in the group of descriptors. Instead they can be used as explanatory variables, which might explain the behaviour of other descriptors.

A *symptom* is a perception, made by means of human observations and measurements (descriptors), which may indicate the presence of one or more faults with a certain probability [ISO 13379, 2003]. By definition there is no symptom without a change in a descriptor value and without symptom these is no fault. Consequentially, a descriptor as such is not necessarily a good fault indicator, but more useful in determination of the baseline data for trending. A symptom descriptive to a change in a descriptor value caused by the fault progress is more useful. We need now to decide, how the change should be defined. It is interesting to know the level of change from normal descriptor value, which can preferably be calculated as an average of the all values obtained during the normal state. In order to get the symptom values, we would subtract the average value from each descriptor value. When the symptom values are defined for classification, it does not really matter, if we use the average values calculated from the total population of the data samples. This is discussed more closely in the next chapter.

Most of the symptoms are typically extracted or derived from a frequency spectrum obtained by a Fast Fourier Transform (FFT) from a time domain data. Other symptom extraction methods have also been presented. Hulkkonen studied the use of a minimum description length (MDL) principle in the novelty detection. She used time domain data from faulty bearings in her experiment and found that it is possible to classify with a reasonably high confidence level various bearing fault modes using MDL directly from the time domain and without performing a FFT. [Hulkkonen, 2008]

A *syndrome* is a group of signs or symptoms that collectively indicate or characterize an abnormal condition [ISO 13379, 2003]. The definition leads to a conclusion that a fault while being an abnormal condition can only be detected and identified through several simultaneous symptoms. This fact is well known, even if often neglected. For one of the simplest faults to identify, imbalance, we can name several symptoms. Some of the symptoms have to be, some might be and some should not be present. Typical symptoms of pure static imbalance are:

- radial vibration high
- axial vibration usually low
- vibration high at rotating speed
- phase angle between horizontal and vertical vibration 90 degrees
- no other harmonic vibration
- no non-harmonic vibration
- no sub-harmonic vibration

As the fault is more complex or there are several simultaneous fault modes, the syndrome indicating an abnormal condition becomes more complex and difficult to interpret. Many of the symptoms listed above, especially high vibration at rotating speed, are common for numerous fault modes. Therefore faults cannot be reliably identified by separate symptoms only, but by the syndromes.

2.5. Diagnostics

Diagnostics are defined as an examination of symptoms and syndromes to determine the nature of faults or failures (kind, situation, extent). The result of diagnostics is a diagnosis. In common language diagnosis, however, has a double meaning; both the actual process and the result. Diagnostics are typically started only after an anomaly has been detected. It may also comprise of additional alternative and supportive measurements and observations. Diagnostics rely partly on known rules that have been well documented [PSK 5707, 2011]. It also involves the inclusion of potential and exclusion of unlikely alternative fault modes and may result in several alternative fault modes with various probabilities.

It may often be difficult to differentiate between a fault and a deviation. For instance, there is always some degree of imbalance in a rotating machine. Standard ISO 1940 [2003] gives balance quality grades for residual imbalance. If the residual imbalance is below the allowable mass, it can be deemed as a deviation, otherwise as a fault. Similar approach can be taken in the case of misalignment, where other standards are applicable [PSK 8301, 2007]. However, for the most other fault modes, it is not possible to define exact fault and failure levels. For instance, a bearing fault typically begins as a metal to metal contact, which might originally be a consequence of an overload or lubrication failure. If the bearing was dismantled at this time, there would probably be no visual sign of a bearing fault. Removal of the primary cause would probably save the bearing from failing. For various other faults definition of a fault and failure threshold is equally difficult.

2.6. Prognostics

Prognostics are defined as an analysis of the symptoms of faults to predict future condition and remaining useful life [ISO 13372, 2004]. Fault progression is a characterization of the change in severity of a fault over time. Prognosis is the result of prognostics process and an estimation of time to failure and risk for one or more existing and future failure modes, and is normally intuitive and based on experience. Prognostics are usually effective for faults and failure modes with known, age-related, or progressive deterioration characteristics, the simplest of which is linear. Prognostics are most difficult for random failure modes [ISO 13381-1, 2004].

Prognostics should not be based on the past and present data only. There are many other factors, such as load or lubrication that have an impact on the fault progression. When an anomaly is detected, it is advisable to shorten the measurement interval to gain more precise information on the symptom progression. The main objective of prognostics is to determine the need of maintenance at a specific time in future. This moment is typically during a pre-defined shutdown. The failure level has been intellectually determined to present an amplitude value of a descriptor or symptom, which would lead to an immediate failure or to a high probability of failure, if exceeded.

Figure 6 presents a situation, where the failure limit is predicted to be exceeded before the next scheduled shutdown and the scheduled shutdown should be advanced to avoid machine failure and its consequences. Usually the measurement interval shall be shortened to give a more precise comprehension on the fault progression. As a result, the time to failure may change drastically.

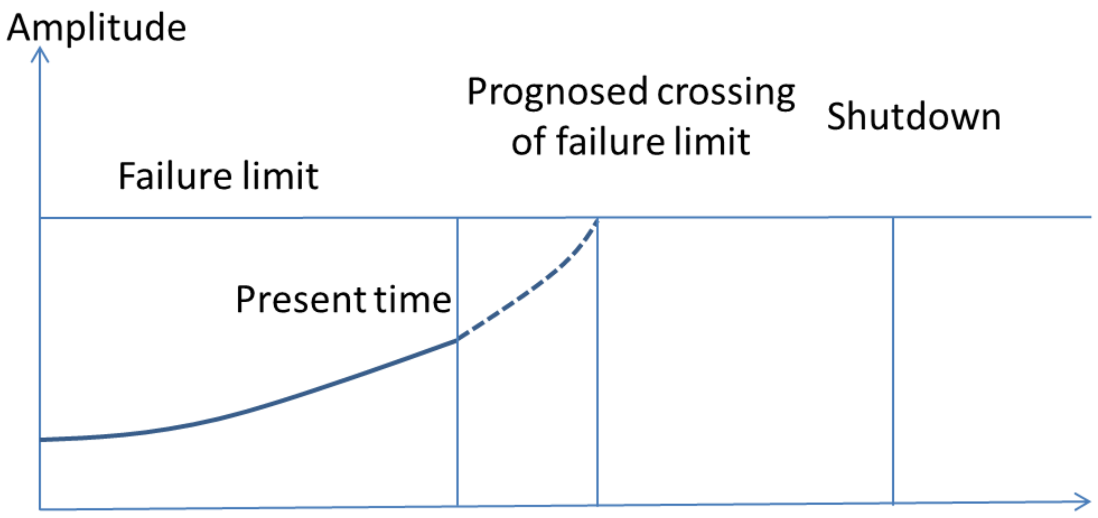

Figure 6: Shutdown needs to be advanced, because the failure limit is prognosed to be crossed before the scheduled shutdown.

Figure 7 shows a situation, where the failure limit is predicted to be exceeded before the second next shutdown. In this case the fault progression should be stopped by maintenance actions during the next shutdown.

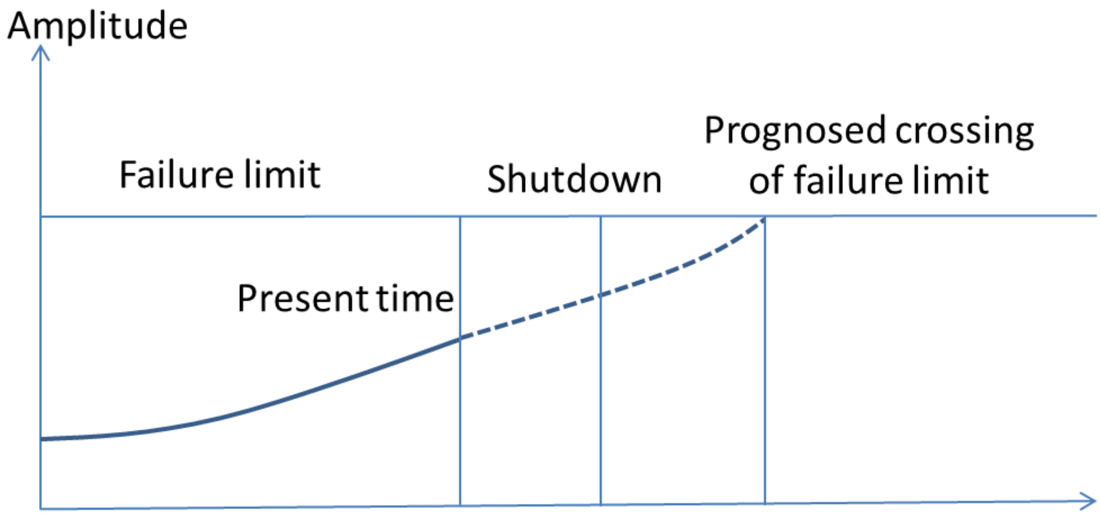

Figure 7: Maintenance is needed during next scheduled shutdown, because the failure limit is prognosed to be crossed before the second next scheduled shutdown.

Figure 8 displays a situation, where the failure level is not predicted to be exceeded before the second next shutdown. Now immediate maintenance actions are not necessary. Other circumstances, such as the increased risk of failure, secondary damages, economical matter, etc. could also have an effect on the timing of a maintenance shutdown.

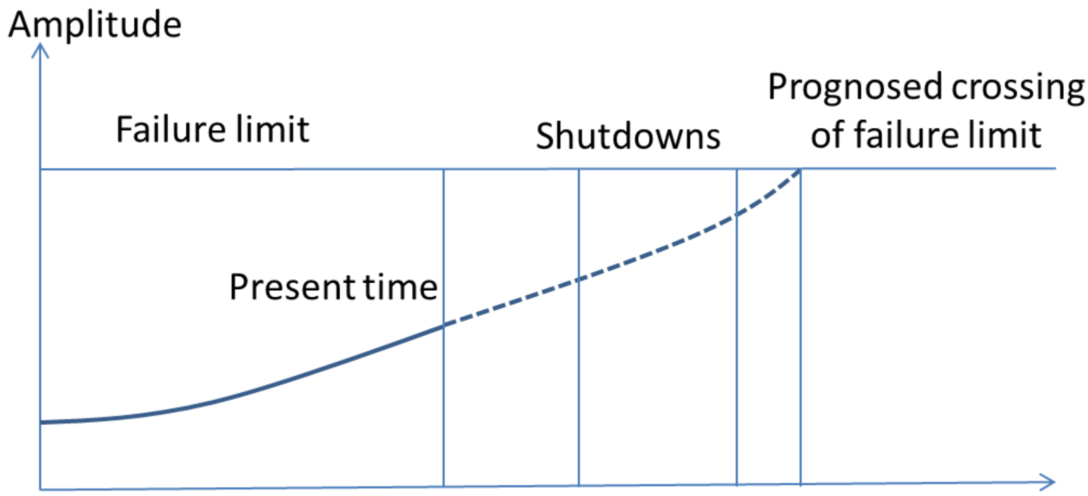

Figure 8: No maintenance needed in the next scheduled shutdown, because the failure limit is not prognosed to be crossed before the next two scheduled shutdowns.

2.7. Prediction

In condition monitoring, prediction is often understood as almost a synonym to prognostics [ISO 13381-1, 2004]. In the context of learning systems, prediction refers to a system's ability to return a label or an interpretation of unknown data based on the introduction of several known samples. The confidence of prediction is highly dependent on the similarity between the known and unknown data and can be evaluated, when the interpretation is verified by other means. According to this definition, prediction means the determination of a nature of a fault, which in fact is the definition of diagnostics. In brain science, prediction means that neurons involved in sensing become active in advance of them actually receiving sensory input. When the sensory input does arrive, it is compared with what was expected [Hawkins, 2004]. In condition monitoring, this would mean that a "smart" sensor has an expectation of a signal to arrive. Prediction is understood in this thesis as the determination of both the fault mode and severity at any given present or future time by utilizing past data.

2.8. Intelligence in prediction

We hear the word smart intelligence in many contexts, such as intelligent maintenance, intelligent machine, etc. However the meaning of these adjectives is often ambiguous. An intelligent system is sometimes understood to have an ability to collect and save performance and predictive diagnostics data. In other applications, it may mean that an intelligent system is able to process this data using rules and pre-set alert limits. In a prediction application, the system should really be able to do more than this. The question therefore is: "Can artificial intelligence be used to predict the condition of a machine at a specified moment of time?"

Before discussing artificial intelligence, let's reason, how a human analyst predicts the machine condition. The diagnostic process is all about knowledge. Knowledge of machines, fault modes and symptoms is widely published [Mikkonen, 2009]. This kind of information is typically theoretical in nature and as such difficult to be applied directly for prediction purposes. It needs to be interpreted and as a result a true opinion or belief is achieved. According to Chisholm, adequate evidence when added to true opinion yields knowledge [Chisholm, 1966]. Adequate evidence can sometimes be learnt only through machine autopsy during the shutdown. On the other hand it is possible to add adequate evidence to true belief without obtaining knowledge. One must therefore carefully distinguish between knowledge and belief. This is an epistemic problem, which has been discussed in detail by Hintikka [2004].

When an expert predicts the current machine condition, he or she attempts to compare the recent data with earlier data from the same or similar machines. The comparison may lead to recognition of an anomaly or a known fault mode. This process is in fact related to pattern recognition, where the pattern is a spectrum or a time domain. If an expert has no previous expertise on a particular fault mode and the pattern is missing, it is extremely difficult to reach a correct prediction. When an expert gets familiar with the connection between the symptoms and the fault modes, he or she understands the relationship between the fault modes and the patterns. It is equally important to memorize and retrace previous interpretations on occurrences of various fault modes.

A human brain has a remarkable ability to use invariant representations and pattern completion [Hawkins, 2004]. Invariant representation means that the brain can adapt to small variances in the data. For instance everyone can recognize a familiar melody even, if it is played with a different instrument or rendition. The brain can also fill in missing data. For instance a human being can easily recognize a familiar face even, if it is seen only partially or from a different angle of view.

In condition monitoring, diagnostic charts illustrating various fault modes are usually given without absolute symptom values. The analyst looks at the patterns in the charts to recognize fault modes. Absolute symptom levels are usually not needed. Artificial intelligence by its nature is likely to fail in prediction, if some data values are missing. Numerical symptom values are necessary, because artificial intelligence is based on computing. Numerical values are also needed to recognize patterns in data. However, the absolute data values might not be important, but the relative values.

The previous discussion suggests that human intelligence relies on the ability to learn, remember and retrace from memory. This is also true for artificial intelligence when using classification as a tool for prediction. It is essential to collect a reasonable amount of machine condition related data for training a classifier. Due to the variation in the operation and environment, the data will vary between sequential measurements. The main objective of a classifier is to organize classes that represent closely similar data by creating classification rules. This allows pattern recognition regardless of small variations in data. The prediction returns either anomaly detection or a labelled class, which may refer to a known normal or fault mode.

An anomaly represents data that has not yet been presented to the classifier. As a comparison to the human intelligence, it resembles an unfamiliar behaviour of a monitored machine. An analyst can further analyse the data or later through machine autopsy gain exact information in order to conclude, if the anomaly was caused by another normal mode or a new fault mode. The classifier has no means to gain more information or make conclusions on anomalies by itself. An analyst's conclusion is needed to label the anomaly data or class. However, a classifier can learn and memorize the relationship between the anomaly data and class. When having been trained, the classifier can effectively predict the correct class for any unlabelled but similar data in future.

The use of intelligent systems in prediction of machine or process condition has been widely studied already for a few decades. Sorsa et all presented several alternative neural network architectures for process fault diagnosis. The study expressed that ten typical process faults could be correctly identified using the multilayer perception network [Sorsa, 1991]. Pöyhönen demonstrates in her thesis, how the support vector machine method can be used in induction motor fault diagnosis [Pöyhönen, 2004]. Wong et all studied a modified self-organizing map for automated novelty detection. The result of this study shows a high degree of accuracy in detecting anomalies. [Wong, 2005]. Yang et all propose a new neural network for fault diagnosis of rotating machines. The method carries out on-line training without forgetting previously trained patterns. The study gave high classification success rates in the fault diagnosis of rotating machinery [Yang, 2004]. More recently, He et all proposes the principal component analysis (PCA) technique in characterizing machine conditions. The experimental results show that the method is effective [He, 2009].

Lee proceeds even further with the intelligence in plant and machine maintenance. The general objective is to perform a "predict and prevent" practice instead of a "fail and fix" operation. This is accomplished by utilizing prognostics and health management (PHM) disciplines that evaluate the reliability of a system within its actual life-cycle conditions in order to detect beforehand any upcoming failures and reduce risks. This approach aims to develop self-maintenance systems and engineering immune systems. [Lee, 2011]

2.9. Confidence level of prediction

The prediction of machine health has many challenges. The traditional way to predict the future condition of a machine relies on the trending of descriptors derived from machine condition data. The trending should indicate, if the machine can be safely operated until the next shutdown or if the shutdown should be advanced. This estimation is based on the pre-defined alarm limits. The setting of alarm limits relies on statistical or historical data. In practice, the limits are imprecise and lead either to too many or too few alarms. Too many alarms result in the frustration of an operator to take necessary actions. Too few alarms might mean that a warning about the deterioration of machine condition is not received early enough.

Even, if an alarm is received early enough, the data might be misdiagnosed. Over-diagnosis occurs when a machine fault is diagnosed correctly, but the diagnosis is irrelevant. A correct diagnosis may be irrelevant, because the severity of the fault is not significant and require no immediate actions. Under-diagnosis occurs as a failure to recognize or correctly diagnose a fault or condition.

The concept of confidence of prediction in condition monitoring has not been accurately determined. The ISO standard 13372 defines the confidence level as an estimate of the likelihood that a calculated reliability will be achieved or bettered. Reliability means a probability that a machine will perform its required functions without failure for a specified time period when used under specified conditions [ISO 13372, 2004]. On the other hand the diagnostic confidence level is a figure of merit that indicates the degree of certainty that the diagnosis is correct. ISO 13379 uses five degrees of certainty: remote, low, moderate and high probability or certain that a given failure mode diagnosis will be accurate [13379, 2003].

The confidence level of prediction could therefore be defined as a degree of certainty that a prediction is correct. We might for instance give a statement that there is a high probability that an unlabelled data sample belongs to a particular class representing a specific fault mode and therefore has a corresponding label.

In essence, prediction should result in estimated time, when the machine should be repaired or maintained latest, i.e. before the machine breakdown, but not before the scheduled shutdown. On the other hand, the estimated time to failure should be as close to the factual time of failure. The problem is that we do not know the factual time of breakdown in advance. If condition monitoring or scheduled maintenance is successful, we might never learn to know it. Hagmark introduces an interesting approach on this subject. [Hagmark, 2011]. For a particular case, the maintenance actions are a result of one of the competing alternatives (CA, SA or PA):

CA corrective maintenance is carried out after fault recognition and intended to put an item into a state in which it can perform a required function [ISO 13306]

SA scheduled maintenance is carried out in accordance with an established time schedule or established number of units of use [ISO 13306]

PA condition based maintenance is performed as governed by condition monitoring [ISO 13372, 2004]

In case of corrective maintenance, there is no advance warning on the fault, before an item enters a state, where it could no longer perform a required function. On the second option, the maintenance work was initiated based on the pre-determined schedule. The third option means that the timing of the work was solely based on the item's actual condition determined by condition monitoring. These options are the only ones possible and therefore competing. They have different maintenance cost factors. Let us define the following three periods:

T_C Operation time to failure (which leads to the CA)

T_S Operation time to the next SA

T_P Operation time to the estimated PA

These periods determine the sequences of events as shown in Table 1. It should be understood that the shortest period comes first, but the sequence of the other two periods are useful in determination of the confidence of condition monitoring.

Table 1 Sequences of competing maintenance actions

Option	Sequence	Description	Result
1	$T_C < T_P < T_S$	PA was planned in vain, because the failure occurred first.	CA
2	$T_C < T_S < T_P$	PA was not planned, but the failure occurred before SA.	CA
3	$T_S < T_P < T_C$	PA was not planned and the item survived until SA.	SA
4	$T_S < T_C < T_P$	PA was not planned and the item survived until SA.	SA
5	$T_P < T_C < T_S$	PA was performed successfully before CA.	PA
6	$T_P < T_S < T_C$	PA was performed unnecessarily, because the item could have been used until SA.	PA

As PA is based on condition monitoring, its confidence level can be evaluated using Table 1. In short, condition monitoring has failed drastically in options 1 and 2. The success of condition monitoring in options 3 and 4 cannot be determined, because we cannot really know, which of the periods T_P or T_C would follow next. The failure has been already been corrected during shutdown. Strictly, we would know T_C only, when it comes first in sequence. In other cases T_C is unknown. We can, however, conclude that condition monitoring has not failed in these options. In option 5, the condition monitoring was definitely successful. The PA was premature in option 6, but again T_C is unknown and therefore the sequence is only theoretical.

Let us leave out the maintenance option SM for a while. The objective of condition monitoring is to perform PA as close as possible to CA, but before CA, i.e. $T_P < T_C$. However, T_C is not known in advance. We can make a natural assumption that the dimensionless relationship $S = T_P/T_C$ is independent of T_C. This can be understood so that the shorter time to failure (T_C), the more accurate is T_P and vice versa. The previous objective is summarized in the following equation:

$$1 - \varepsilon \leq S < 1 \qquad (2)$$

where $\varepsilon > 0$.

In fact, *S* as an independent random variable now represents the confidence of prediction being correct. If S is close to but less than 1.0, the timing of PA is optimal. Figure 9 shows two examples of distributions of variable *S*. The solid line marks a distribution, where PA will happen before machine breakdown (CA), but in most cases far too early. The solid line represents a distribution, where PA will often happen in time and seldom too early, which is usually preferable.

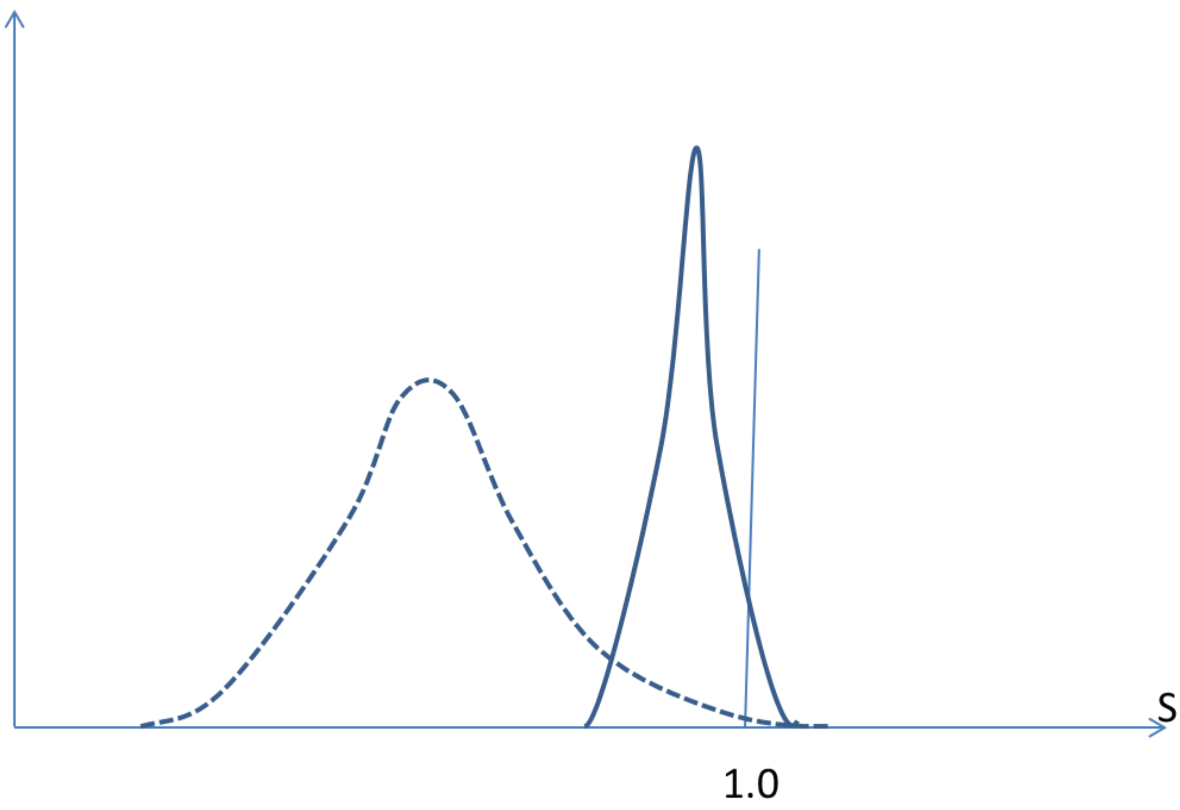

Figure 9: Distributions of S (T_P/T_C) illustrates that the prediction of the timing of condition based maintenance becomes more accurate, when time to failure is close.

Hagmark [2011] has further simulated the operation periods T_P, T_S and T_C and costs of maintenance. These models give a good comprehension on the effect of various competitive maintenance actions. The usage of real data yields excellent information for decision making.

2.10. Certification

The confidence level of an analyst's ability to interpret machine condition related data can be tested and verified by a certification process. ISO standard [18436-1, 2004] gives the general requirements for training and certification of personnel performing condition monitoring and diagnostics of machines. Specific requirements for personnel performing vibration condition monitoring and diagnostics are presented in ISO standard [18436-2, 2003]. Certification is offered by at least the following organizations: Mobius Institute Board of Certification (MIBoC), British

Institute of Non Destructive Testing (BINDT), Vibration Institute (VI) and Technical Associates of Charlotte (TAC). If the certification body has been accredited, the certificates are acknowledged by other bodies and around the world.

To be eligible for certification the candidates shall have a combination of education, training and experience to ensure that they understand the principles and procedures applicable to the machinery condition monitoring and diagnosis technology. Each technology is divided into three or four categories. For each certification category, the candidates shall be required to answer a number of questions over a specified time. After the candidate satisfies all requirements for certification at a given category, the certification body should announce the certification, and issue certificates indicating certification. The period of validity shall not exceed 5 years from the date of certification indicated on the certificate.

The ISO standards also include the code of ethics, which will further increase the confidence level of condition monitoring and diagnostics. At the option of the certification body, after reviewing evidence of unethical behaviour, the certification shall become invalid.

3. DATA PRE-PROCESSING

3.1. Challenges

There are several general challenges that need to be considered in data analysis before inputting data into the classification. These include missing or erroneous data, missing descriptor values, irrelevant data, random values, outliers etc. The data analysis may lead to wrong conclusions, if these aspects are not recognized and handled properly. [Berthold, 2010]

An additional concern is the presentation of descriptors in several different units. In conditioning monitoring applications, we may, for instance, wish to analyse vibration velocity and acceleration values or vibration and process values simultaneously. Some descriptor values may vary between 0.1 and 5 and others between 20 and 100. Without data pre-processing the former values will probably lose their significance in the analysis. In order take advantage of all descriptor values the data should be normalized.

By definition a symptom is a perception made by means of features (descriptors), which may indicate the presence of one or more faults with a certain probability. In other word, if a feature value stays stable, it does not indicate a presence of a fault and there is no symptom.

One of the goals of this thesis is to demonstrate the generalization of data collected from different items. This would allow use the data related to a specific fault in a single machine to benefit the other machines in comparable conditions. Machines are individual and the descriptor values may not be same, even if the conditions are equivalent. Use of symptom values instead of descriptor values will likely solve this problem.

3.2. Missing data

In an ideal situation, a complete set of data including all faults in various progressions for all machines is available. This way the system could be trained to identify all potential states. In practice, this is impossible and especially fault data will typically be missing in the beginning. Also, it is more than likely that all of the normal states cannot be experienced during the initial training period. Due to the ability to retrain the system, this is not a major problem. Whenever data that was initially missing is tested with the system, it will appear as an anomaly and as such will be drawn attention to.

Some of the descriptor values might be missing from the data set because of various reasons. It is often not desirable to discard the whole data sample because of this. These values should never be replaced by zero. This would result in errors during data analysis, because zero is a significant value. A computer software allows inserting a Not a Number value (NaN), if a descriptor value is not available. A NaN value causes no operations to be performed for the variable. In this way the missing value can be predicted, as the classifier has a known weight value for this variable.

3.3. Erroneous data

Errors in data might be generated during the data acquisition and collection. The handling of erroneous data is difficult. In fact, the neural network system as such has no means to detect errors in the data. During training, the errors will have an impact on the classifier and may distort it. During prediction, the errors may, however, cause anomaly detection and would be investigated closely. Data values should be evaluated during the pre-processing. Any abnormal set of data should be discarded from further processing. In a condition monitoring application, the changes in data values between successive measurements are typically small. Even during a fault progression, the descriptor values change slowly. If a significant change or deviation is detected, a re-measurement could perhaps be taken. If the change is not permanent, the data should be discarded. On the other hand, if erroneous data is symptomatic of a fault in the measurement or monitoring system, it could perhaps be reported as an error.

3.4. Outliers

Outlier is an observation that is numerically distant from the rest of the data. In condition monitoring, the data value is typically repeatable, when collected in comparable situations at the same location. An outlier deviates significantly from the successive values in trending. In such case, the measurement should be re-collected or rejected. During training, an outlier might still be detected by investigating the number of data samples used to train a class. If a class has only a few training samples and they have a great variation, it is probable that one of the samples is an outlier. An outlier will distort one or several class centres and borders and should therefore be removed from the training data set. During prediction, outliers can be detected as anomalies.

3.5. Precision of data values

The values may be imprecise for several reasons. Some of the values may suffer from poor dynamic range. This causes low amplitude values to fluctuate relative strongly from the average value regardless of the machine condition. A small change in the actual value may cause a significant change in the data value. This may result in unexpected classification results depending on the type of normalization process.

It may appear tempting to use as accurate values as possible. However, the descriptors should in the first place be selected so that a minor deterioration in machine condition causes a major change in the descriptor and thus in the symptom value. Therefore small variation in the symptom values can be considered to be caused by other phenomena that the machine operation. The training of the classifier attempts to organize the classes according to the variation in the data sets. In this case this might result in inadvertent diversity of classes. Threshold setting and rounding can be used to equalize the insignificant data variation. In the prediction process the accuracy of data has no impact.

3.6. Irrelevant data

In some cases, the collected data may include samples that are not related to the current condition of a machine. In particular, this happens in low speed or load conditions, but the data may also have been collected at a location, where the vibration response does not correlate with the forces induced in the machine. A classifier can handle irrelevant data, but the classifier will be unnecessarily occupied with insignificant classes. Therefore it is advisable to discard all such data samples from the training and prediction data.

It might be difficult to know, if data is relevant until all possible faults have been encountered. Preferably, a chosen descriptor should be orthogonal, i.e. a symptom does not cause similar changes in more than one descriptor.

It is also possible that the training data includes data sets that represent an operation condition that is unlikely to happen again or for other reasons does not need to be predicted in future. This could perhaps consist of data collected during an exceptional operation of a machine or malfunction of the measurement system. Depending on the selected symptoms and the nature of the phenomenon, such data typically is grouped by the classifier in one or a few classes. Therefore they are easily identifiable and should be removed from the training data.

3.7. Normalization

The purpose of normalisation is to isolate statistical error in measured data by making it commensurable. Normalization refers to the division of multiple sets of data by a common variable in order to negate that variable's effect on the data, thus allowing underlying characteristics of the data sets to be compared. This allows data on different scales to be compared, by bringing them to a common scale.

Normalization offers solutions for several concerns:

- values in different units made commensurable
- values with different ranges of change
- symptom extraction from descriptor values
- generalization of data from various sources

Berthold & Hand advice to use the standard score method with principal component analysis [Berthold, 2003]. In statistics, a standard score indicates, how many standard deviations an observation or datum is above or below the mean. It is a dimensionless quantity derived by subtracting the population mean (μ) from an individual raw score (*x*) and then dividing the difference by the population standard deviation (σ). The data can be normalized with a standard score method using the following formula.

$$z = \frac{x - \mu}{\sigma} \qquad (3)$$

The outcome of normalization in fact makes data values commensurable by being dimensionless and equalizing the ranges of changes. With the centring of the variables, the original descriptor value x is transformed to a symptom value z. Standard score is also called z-score [Berthold, 2010].

4. SOM BASED CLASSIFICATION

4.1. Introduction

Classification can be defined simply as a process responding to a question: “Which group of known samples does the new unknown sample belong to?” Classification is a process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects, whose class label is unknown. The derived model is based on the analysis of a set of training data. [Han, 2001].

For presentation purposes, most of the figures in this chapter show data in two dimensions representing two symptoms used to define the condition of a machine. In reality, the presentation shall be extended to a multi-dimensional hyperspace. The number of dimensions represents the number of descriptors or symptoms used. Depending on the object under investigation and on the potential fault modes the number of symptoms might be more than twenty or even over one hundred.

Various classifiers are available for prediction, but their differences are not discussed in this chapter. There are, however, two general categories of classifiers: supervised and unsupervised. A supervised learning algorithm requires that the output, also referred to as a label or an interpretation, of the training data is known. The creation of classes is based both on the similarity of the training data samples and their output. In unsupervised classification, the classes are created entirely based on the similarity relations of the data samples. Therefore, unsupervised classification can be considered primarily as a clustering method [Kohonen, 2001]. The main reason for selecting an unsupervised classifier for this study is that in condition monitoring applications the output or interpretation of the training data is typically not known at the time of training the classifier.

This study refers to classification as a tool to detect normal modes, anomalies and faults. This is accomplished by the creation of classification models based on the rules derived from the training data samples and labelled by an analyst. Classification is used to predict the class label of any new yet unknown data sample. When the classification model is accurate, the class label can be predicted on a high confidence level.

4.2. Self-Organizing Map

The main objective of using a Self-Organizing Map (SOM) classifier is to predict the class of an unclassified data sample on the basis of known training samples. Being an unsupervised learning method SOM is basically a clustering tool. Clustering tries to group a set of data samples and find the relationship between the samples. Therefore, as such SOM does not fulfil the definition of a classifier. When the groups or classes are labelled, the classification model has been created and SOM can be used as a classifier.

The following sections present the various operations, where classification is used as a part of a continuous learning. A self-organizing map (SOM) is used to create the classifier and it provides the following information for each of the trained classes:

- class reference (number, label)
- number of samples used to train the class
- Euclidean distance of the furthest sample from the weight centre (maximum distance)
- weight centre as a set of symptom weights
- minimum symptom values in the class for each symptom
- maximum symptom values in the class for each symptom

The class reference is used as a pointer to a specific class in the classifier. It should also include the label to indicate the fault mode and severity.

The basic idea of the Self-Organizing Map is to capture the multivariate distribution of data samples by creating a low dimensional latent surface on which the data samples are projected. The latent surface is approximated with a lattice of units, neurons. In most applications the dimension of the SOM is two (surface) to make the visualization of the mode easy [Lensu, 2002].

The Tree-Structured Self-Organizing Map (TS-SOM) has some special characteristics. The classifier map consists of selectable number of layers, which define the number of classes used. See Figure 10. Each layer has $2^{2(l-1)}$ neurons or classes, where l is the layer index. For instance, the third layer has $2^4 = 16$ neurons to represent the data. The top layer has been trained with the total sample population and has only one class. Each class in the tree-structure has four children, who inherit the initial weight values partly from their parent and partly from the parent's siblings. Training is an iterative process that will be repeated to reach the pre-defined allowable error level.

The required number of classes depends mainly on the divergence of data and the required accuracy of prediction. In a condition monitoring application, there should be enough classes to correlate to the number of various fault modes and severities. One of the disadvantages of SOM is that the number of neurons has to be determined in advance. This may result in situations, where only a small amount of groups represent models. The remaining groups have been created in vain.

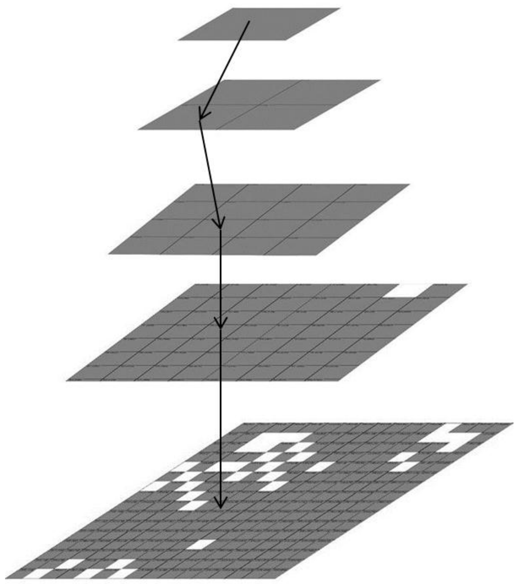

Figure 10: Tree-structured Self-Organizing Map illustrates that the search process is started from top layer and the best matching class is searched among the four sons of the class. The process is continued accordingly until the lowest layer has been reached.

Using the tree search through the higher (smaller layer index) SOM layers, the correct class of the current layer can be found without performing the full search of all neurons. Therefore, TS-SOM improves the training in its most critical step and reduces the complexity of the search [Lensu, 2002]. The improvement offered by the TS-SOM algorithm is no longer equally important, because of an increased processing power in modern computers. TS-SOM method is used in this study because of its illustrative features.

The map is organized during training in a way that the neighbouring units process similar representations. This resembles in fact the way brain maps the neurons to economize communication [Churchland, 1996]. Any class that has not learned from even a single sample is left blank to demonstrate that it will not be used in the prediction.

4.3. Training of SOM based classifier

The training algorithm attempts to organize all similar data samples into a same group. The similarity is typically measured by the Euclidean distance between the data samples. During the training, the data samples adjust the weight vector coordinates through an iterative process towards the weight centre of all data samples used for training of a particular class. See Figure 11 [Lumme, 2012a]. Note that the iterative training process is not complete and the weight centre is not yet at the centroid of the training samples.

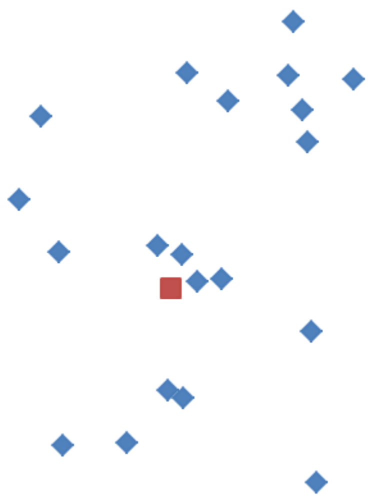

Figure 11: Weight centre (square) achieved through an iterative adjustment by training samples (diamonds).

After the training, the classifier consists of a set of groups that become classes, as soon as they have a label. Each class has a characteristic weight vector, which will be used to search for the best matching class for any new data. Note that SOM is an iterative process that may be terminated before the weight vector is in the centroid of the data samples. A classifier that was trained using supervised learning will have all classes labelled. For a classifier trained with unsupervised learning, the classes should be labelled manually. Typically, in the initial training phase all class labels should present a normal mode only, unless a known fault mode has been included in the training data set. In such a case, the classes that were trained using these data samples should be labelled accordingly.

Once a set of classes has been created, the label of any new data can be predicted by localizing the best matching class. Figure 12 shows an example on a set of ten classes with their weight vectors. The lines mark the approximate borders between the classes. Together they form the Voronoi tessellation [Kohonen, 2001]. All data vector within borders will have the shortest distance to the same weight centre and therefore will have the same label. Clearly, the presentation is much more complicated in a multi-dimensional hyperspace, where the hyper-planes are used instead of border lines.

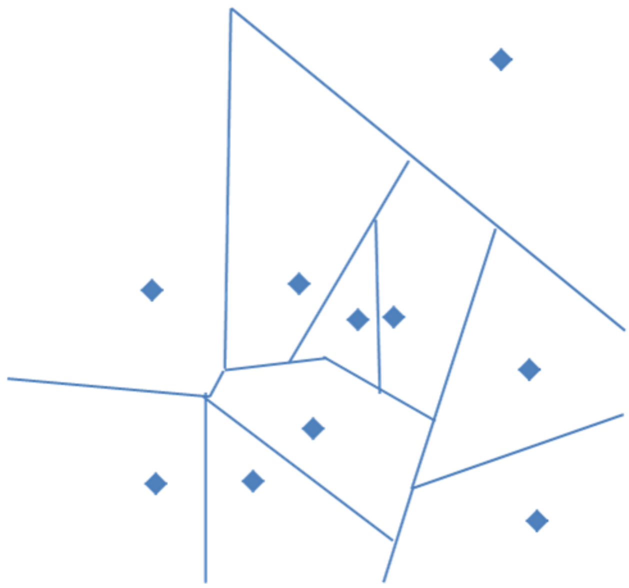

Figure 12: Class borders that have been determined by the shortest distances to the weight centres in several classes.

Figure 12 suggests that for any new data sample a best matching class can always be found by locating a respective weight centre with a minimum distance to the data sample. However, one should keep in mind that the new data sample might still locate at a distance, which is rather far from the weight centre of the best matching class. Such a data sample does not belong to any group of data sets used to train the classifier. A border has to be drawn somewhere in the hyperspace around the weight vector to detect anomalies.

In Figure 11, the samples that are close to the weight vector have a great membership within the class. Theoretically, statistical distribution can be determined using the training data. The distribution can be illustrated by using contours as in Figure 13. In a practical application, the distribution itself might not be of interest. Preferably, we are really interested in the border of the class. This goal can be reached by selecting a suitable membership contour to a class border. During testing, all samples falling outside of the contour would be considered as anomalies.

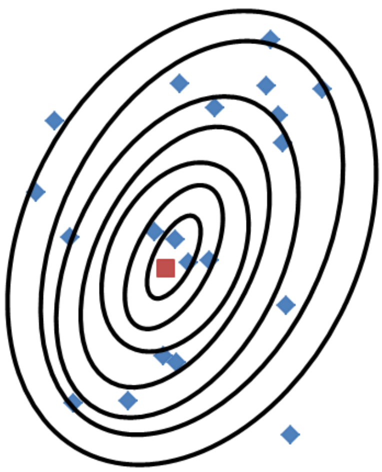

Figure 13: Statistical borders that are based on a true distribution of the training samples (diamonds) around the weight centre (square).

In Figure 14: a contour has been drawn through the furthest data vectors. Such a contour would probably be an excellent class border in a hyperspace, but it is difficult to implement. Another solution for a class border is the usage of a rectangle in a hyperspace. The shape and size of such a rectangle shall be defined by the minimum and maximum symptom values in the data set.

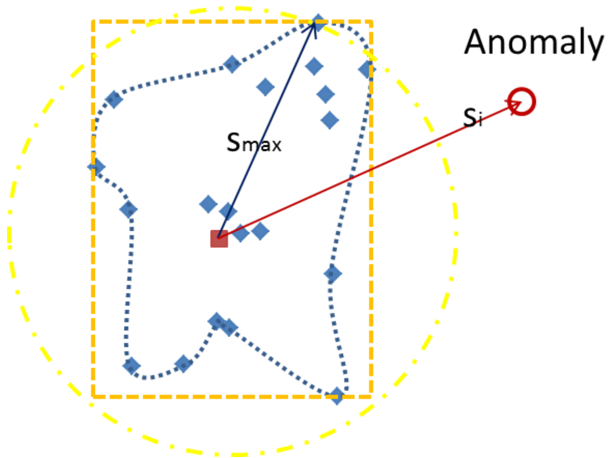

Figure 14: Various options to define the class borders. The orange rectangle defines the borders based on the maximum values in each dimension. The yellow circle illustrates a spherical border with a radius at the longest distance (s_{max}) of all data samples (diamonds) from the weight centre (small square). The blue curve runs through all the local maximum distances to produce a contour type border. The distance (S_i) of any new

data sample (red circle) is compared against the class borders to define its status. If the data sample falls outside the borders, it is considered an anomaly.

As an easiest method, one could use the hypersphere with the radius at the maximum distance of all training data samples and centred at the weight vector. Any new data vector falling within the hypersphere or close to it would have the same class label. Any data vector clearly outside the hypersphere represents an anomaly meaning that a similar data has not yet been presented to the classifier. Data vectors close to the surface of the hypersphere might belong to the same class at some confidence level or are anomalies. A closer analysis at the data should reveal, if the data should have the same label. For the purpose of clarity and illustration, this study does not rely on class borders, but on class membership and anomaly index instead.

4.4. Class membership

Class membership can be used to evaluate the confidence level of diagnosis. When interpreting a new sample, a classifier looks for a best matching class, which obviously will always be found. When comparing the new sample with the samples used to train the class, an assessment can be made on the membership within the class. If the new sample falls close to the weight centre of the training samples, the membership is high. If on the other hand, the sample is further away from the centre than any of the training samples, the membership is low. Taken that the classifier has been accurately labelled, the confidence level of diagnosis would then be high or low consequentially.

Membership functions are typically used in fuzzy logic, but let's define a useful function for classification purposes. We may take a linear approach by defining that we have a maximum (1.0) membership (confidence of diagnosis) at the centre and a pre-defined (for instance 0.5) membership at the maximum distance. This can be expressed as follows.

$$y = \max(1 - c\,\frac{s}{s_{max}}; 0) \qquad (4)$$

where y is the membership value, c is the membership constant, s is the new sample's Euclidean distance from median and s_{max} is the Euclidean distance of furthest training sample from the weight centre.

When s is zero, i.e. the sample is at the centre, the membership value will be 1. When s is s_{max}, the function will return 1-c as the membership value. For any distances greater than s_{max}/c, the membership will be zero. A value of 0.5 is recommended for the membership constant. This will result in a 0.5 membership at s_{max} and zero membership at distances longer than two times s_{max}.

A linear model is not ideal, because it results in reasonably high membership values at long distances from the centre and low membership values quite close to the centre. Instead a sigmoid function expressed in the following equations is more appropriate.

$$y\,(t) = \frac{1}{1 + e^{-t}} \tag{5}$$

$$t = k \times (1 - \frac{s}{s_{max}}) \tag{6}$$

where k is a skew factor, s the distance from median and the s_{max} the maximum distance from centre.

A skew factor k of 5 would yield a membership function given in Figure 15. The function gives membership value 0.99 at the centre, 0.5 at the maximum distance and 0.01 at two times the maximum distance.

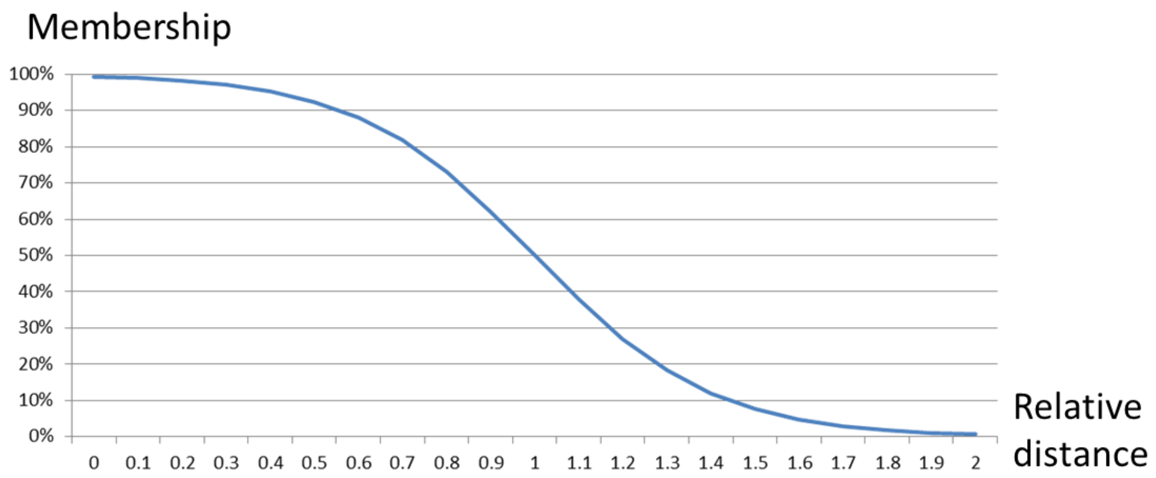

Figure 15: Sigmoid function gives high membership values, when a sample falls close to the weight centre (relative distance is close to zero) and low membership values, when a sample resides further from the weight centre than the furthest training sample (relative distance > 1).

It is usual in condition monitoring applications to display any anomalies and fault progressions as increasing values. Therefore, this study uses an anomaly index instead of a class membership. An anomaly index (AI) can be defined as a ratio of a distance (*s)* of the data sample to weight centre to a distance (s_{max}) of the furthest training sample to the weight centre.

$$AI = \frac{s}{s_{max}} \qquad (7)$$

4.5. Multivariate classifier

The previous figures used two-dimensional presentations of data for simplicity. Figure 16 highlights the complexity of a classifier in three dimensions. Additional dimensions are difficult to be shown on a two-dimensional paper. The numerous spheres represent the various classes created through training. The radiuses of the spheres represent the distances of the furthest training data from the sphere's centre point. Some of them form clusters and might intersect. However, a hyperplane can always be found, which separates the two close classes from each other. On the other hand, the closely neighbouring classes are likely to be representing data with the small variations only. In reality the fault modes will show in a low membership, i.e. in a high anomaly rate, to any class labelled as normal, if the descriptors are correctly selected.

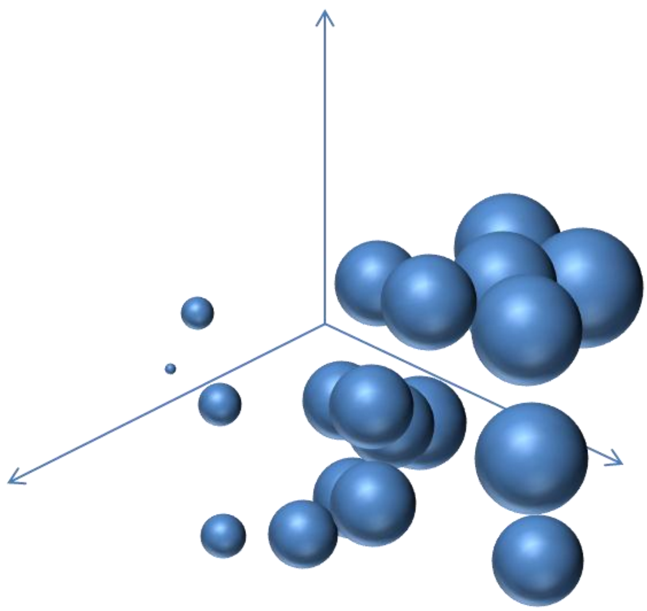

Figure 16: Three dimensional classifier, where the classes are shown as spheres with radius at the maximum distance from the weight centre. Note that the borders of adjacent classes may intersect.

Clusters define a group of classes with close similarity. Empty spaces between the clusters indicate still “unknown classes” that have not been trained with any data samples. These clusters might be difficult to distinguish in a low dimensional presentation.

4.6. Class label

For machine diagnostic purposes, the class label should be as exact and descriptive as possible. For all classes that are related to normal mode without any early indications of beginning fault modes, simple label “normal” or without label may suffice, but for other classes the label should include both the fault modes and severity. One should keep in mind that the syndrome may be caused by more than one fault at a time. Condition monitoring typically reveals first a primary fault, for instance imbalance that causes a secondary fault, such as a bearing failure. Symptoms of both fault modes appear in the data.

The fault severity as a part of class label can be expressed in several ways. Fuzzy definitions, such as “moderate”, “serious” and “extreme” are often considered practical, but prediction of a remaining operational time is more informative. The prediction should include the confidence level, for instance “the machine can be safely operated 30 days at 80 per cent confidence”. As more data becomes available and the time to failure becomes chorter, the confidence level of prognostics becomes more accurate.

A class label could also have a reference to earlier data samples that were used to train this particular class. Often background and post-diagnosis information is available from the previous occurrences of machine condition problems that might be useful in assessing the severity of the current fault mode. This information might also guide to planning the required maintenance actions quicker.

5. CONTINUOUS LEARNING

5.1. Introduction

The objective of continuous learning is to utilize all data to effectively predict the current or future condition of a machine. This is particularly important, when the monitored machine has not yet experienced any problems or is faced with repeating similar problems. Continuous training is started typically, when a new set of similar machines are taken under monitoring. Initially, we may have no condition data and information on the machines at all or it has not yet been utilized. In order to learn the normal condition characteristics of a machine, an initial training needs to be done. It will result in classification models that can be directly used to predict machine condition, if it remains normal. In addition, it will also predict any anomalies, i.e. deviations from the known normal condition.

Anomalies bring along new valuable information. They may be caused by new normal modes not encountered during the initial training or by fault modes. In either case, the related data should be saved and the system retrained to minimize the false alarms caused by known anomalies or react to known faults. This process is continuous and is repeated, whenever new anomaly or even fault data becomes available. The data may originate from the same machine or a substantially similar machine. Generalization of classification models is vital, because it is unlikely that a single machine faces all potential fault modes, but a particular fault mode may have been encountered somewhere else on a similar machine. It is important to predict reliably the current and future machine condition at first instance. This chapter explains the steps needed to implement a continuous learning system.

5.2. Initial Training

Before the initial training, a reasonable number of data samples should be collected and the descriptors derived from the data. The population of the data sets should preferably include data from the various operational modes of the machine, but data can also be added through a retraining process, if such data becomes only later available. Depending on the machine’s operational cycles, the collection of data may take from a few hours to several weeks or months. For statistical reasons, the number of data samples should exceed 100. Typically, data from a machine, when it is running in a fault mode is not available for initial training. This can also be added to the training data later through retraining.

The map in Figure 17 was created on the set of 3534 data samples from a machine in a good condition. A five layer classifier with 256 classes was selected. Before the training operation, all data samples were normalized using the standard score method presented earlier. Due to the initialization algorithm, the map tends to be organized so that the classes with high weight vector norms are located in the top left corner, while those with low weight vector norms are in the opposite corner. The weight vectors in adjacent classes are in the same order of magnitude.

The TS-SOM algorithm used to create the map presentations in this thesis has a specific numbering system for the class indexes. The index number starts on the top layer from zero and leaps through all layers continuously. Therefore, on a five layer map the index number starts from 85. As four adjacent classes are descendants of a single class on a previous layer, they have successive numbering. For instance, the classes in top left corner have indexes 85, 86, 87 and 88. This method results in peculiar numbering in the middle of the map. For instance classes 149 and 234 appear as close neighbours. However, the indexes are just pointers to the classes and the numbering sequence is not important. The numbers in the brackets indicate the amount of data samples used for training the specific class. The more samples used, the more reliable is the training of the classifier.

Some of the classes in the map have not learned from a single data sample. These classes have no value in classification and should be excluded in the search process for the best matching class. They have been marked with white background on the map. Several other "white" classes can be seen on the map. These classes typically separate classes, where the Euclidean distance between the weight centres is significantly long, but untrained classes may also be a result of the training process itself. A unified distance matrix (U-matrix) offers a more reliable presentation of distances between neighbouring classes [Kohonen, 2004].

After the initial training, almost all classes appear to be populated. This is typical, if the initial training data is homogenous. As the classes are always used effectively, it may look like there is no space left for any retraining of anomalies or known fault modes. The classifier will, however, adapt to the data volume, when new data is appended to the training data. This means that smaller number of classes will be used to represent slight deviations between adjacent data samples. On the other hand, in most cases the symptoms should have been selected carefully so that any changes in a machine condition would cause a major change in the symptoms.

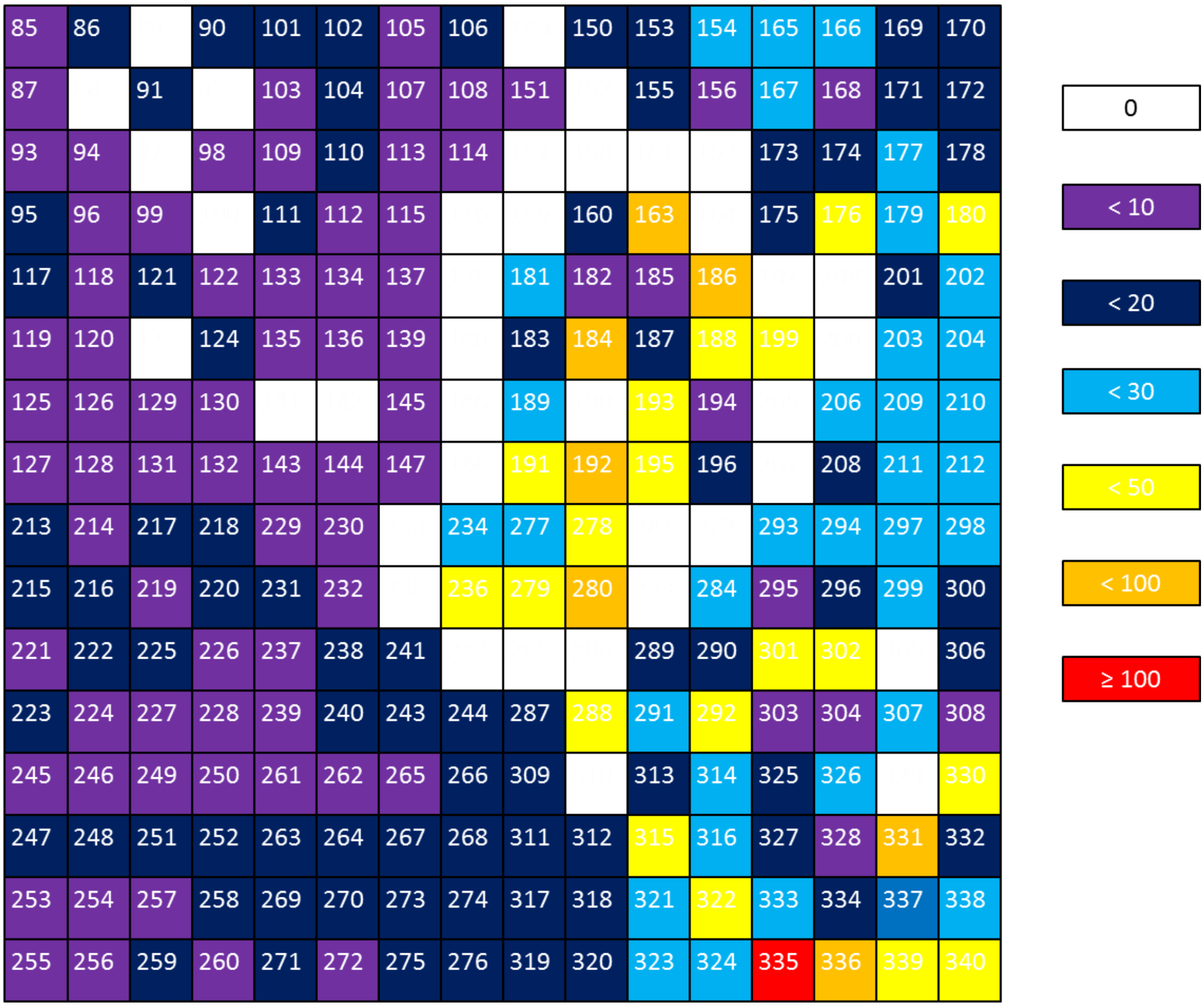

Figure 17: Map after initial training shows the number of training samples in colour codes. Most samples are used to train the classes in the left bottom section.

5.3. Prediction

After the initial training, the classifier is ready to be used for prediction of any new unlabelled sample. Before the actual prediction, the descriptor values should be normalized using the same scale and offset as with the initial training data or alternatively the weight factors of all classes should be de-normalized. During the prediction, the Euclidean distance between the data sample and the weight centre of each class is calculated and the one with the minimum distance shall be selected as the best matching class.

After the initial training, the prediction basically returns information, whether the tested samples have known labels or if they are anomalies. This information can be expressed in a simple method, for instance by using traffic light signals, where green indicates that the object is in a known good condition, yellow that an anomaly has been detected or red that a known fault condition has occurred. The prediction could first be performed on the training data, which should result in an equal amount of hits and training samples in the classes and no anomalies. If this is not a case, the classifier might be inaccurate.

While the borders between classes are defined by the closest distances, a border with an anomaly is a fuzzy definition. If class membership is high, the tested sample belongs to the same class and will have the same label. Low membership represents an anomaly. A class border has to be drawn somewhere.

When an anomaly is detected, it should be interpreted with human expertise. This may involve analysis of raw data, comparison of syndromes or supplementary data to be collected. The interpretation does not need to be completed at the time of detection and not even necessarily before retraining. The samples related to the anomalies should in any case be labelled for reference so that the respective class on a new map after retraining can be found and labelled accordingly. Usually it is not necessary to label classes representing normal modes, as normal mode is considered as the default mode. There might be times, where we would prefer a certain normal mode to be detected. In such a case the respective class can also be labelled.

Data classified from an abnormal condition represents novelties and are therefore necessary to be retained for two reasons. If the data was collected, when an object has experienced a previously unknown normal mode, new similar samples will no longer be diagnosed as novelties. In the case of a known fault mode any new data from the various severities of the fault is valuable in prediction of the fault progression.

5.4. Retraining

Retraining enables the continuous learning process. Retraining is necessary, whenever new data is appended to or removed from the training data. This happens, when new anomalies caused by new normal or new fault modes are encountered. Retraining is also required, when the training parameters are changed.

Before retraining, all data samples should be normalized in the same way than with the initial training. It is recommended to use the whole population including the appended data samples in the calculation of mean and standard deviation, which will clearly be changed.

Also, before any retraining the class interpretations should be memorized. A patented method [Lumme, FI115486] can be used to save both the class labels and weight centres for all classes with labels (known fault modes and unresolved anomalies) and restore the labels to the new classes respectively after retraining. By saving the class labels and weight centres temporarily, their connections will remain unchanged. After the retraining the organization of the classifier will change and the class indexes will no longer be the same. Especially, when the anomaly data differs significantly from the previous training data, classes with new weight vectors will be generated. They will force the previous data samples to be used in the training of fewer classes. In a way, this can be understood as a reduction of the training accuracy, because the weight centres and borders will be affected. On the other hand, the reduction will happen mostly in classes representing similar classes, i.e. normal modes.

After the retraining, the saved weight centres will be used to locate the best matching class on a retrained map. Clearly, the best class will always be found and it shall be labelled with the saved class label. The class has likely been trained by more data samples than during the previous training. Theoretically, some of these training samples could have different interpretations. In case this is undesirable, the classification accuracy should be improved by increasing the layer count and thus the number of available classes. Ultimately, the selection of descriptors should be questioned, if adequate distinction between different modes cannot be detected.

Figure 18 shows the updated classifier after the first retraining, when 200 new data samples classified as anomalies, were appended to the initial data. It can be seen that the map has changed in shape. This is a result of the classifier’s ability to adapt to new data volumes and types of data.

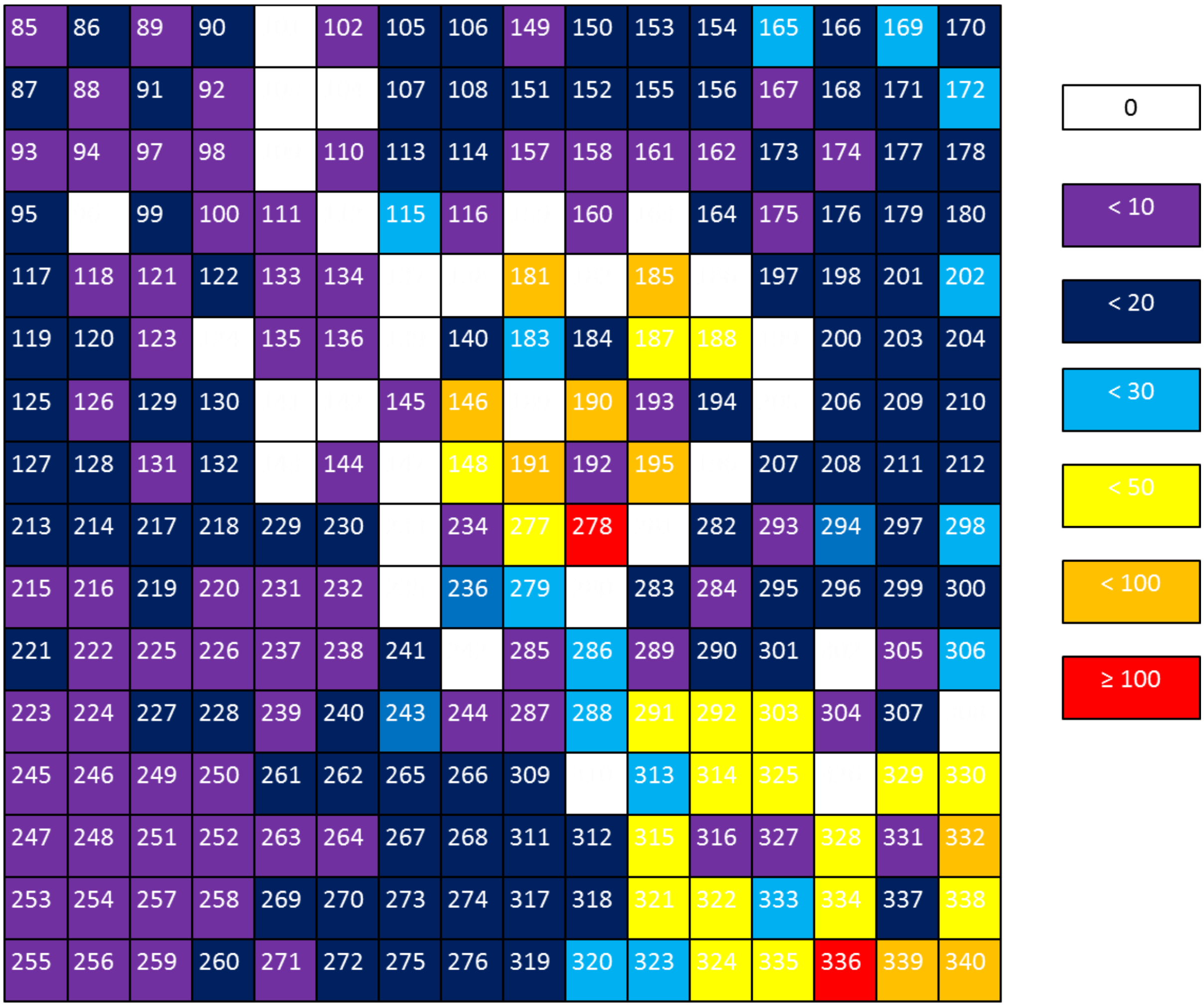

Figure 18: Map after retraining with colour codes indicating the number of training samples in the classes. The classes in the right bottom corner are highly populated.

5.5. Fault Detection

A classifier will not be able to detect any fault modes until it has been trained with a data containing descriptors related to a specific fault mode. This is in fact similar to human expertise. A machine analyst cannot diagnose any faults that he has not experienced nor has prior knowledge about. While an analyst can receive knowledge through experience and education, a classifier will gain knowledge through retraining supported by human expertise. Once an anomaly has been detected and the respective class in the re-trained classifier has been labelled as a fault mode (without anomalies), any new data finding its best match in this class will be automatically interpreted accordingly. The confidence of diagnosis is extremely highly dependent on the accuracy and correctness of the original labelling accomplished by the human analyst. In an ideal case, the analyst would give a precise description of the fault mode and severity including the remaining safe operational time. This interpretation can be based on post-diagnosis, when the problem has been studied in details and uncompromised evidence has been provided to support the diagnosis. For any new sample classified in this group of data samples, the membership within the class will indicate the confidence of diagnosis.

5.6. Fault Progression

In cases, where the best matching class is labelled as one of the fault modes or the predicted data has novelty symptoms, the user should again be alerted. The confidence of diagnosis for this particular fault mode might be poor, but the new sample might have symptoms suggesting that a more severe (or sometimes less severe) mode of the same fault has been detected.

When the fault progresses to a more severe mode, the symptoms will change. It is important to understand that some symptoms may increase and others may decrease in value. Also, in the beginning there might be only a primary fault present, but due to the excessive load caused by it, a secondary fault with its own characteristic symptoms may develop. For these reasons, the syndromes should be studied instead of single symptoms. An early warning of fault progression is received as an appearance of anomalies, as was the case for the first instance of the early fault. If not retrained with new data, the fault progression will be detected as a lower membership within the class.

On user's initiative, the retraining can be started in order to differentiate between the various severity modes of the fault. When the fault progression has ceased, possibly because of maintenance actions or a failure, the full history of the fault progression can be displayed as a trajectory on a map. This path allows predicting, what will probably happen next, when a new instance of a similar fault is first detected.

5.7. Probability of Failure

There are several points that should be taken into consideration, when the probability of a failure is evaluated. First, it is a common assumption that the probability of a failure is proportional to the magnitude of descriptors. In many cases this might be true, but more often the fault progression can be seen as the change in the symptom distribution, i.e. in the syndrome. Depending on the fault mode some symptom values may or may not increase, stay steady or decrease in magnitude. Therefore the estimation of the probability of a failure cannot rely on the magnitude of a single symptom only.

Also, the relationship between the severity of a fault and the symptom magnitude is often not linear. Usually this relationship is not known accurately. Even, if the failure has happened before and the symptoms were recorded, the next occurrence of a same failure mode might not give the same symptom values. It is therefore often considered satisfactory, if we can estimate the probability of a failure at a certain time. This can typically be expressed as a probability that a machine will operate without failure for a given period.

Some additional uncertainty is added, when we take into consideration the ability of the vibration analyst to diagnose the symptoms to the fault modes and severity. The symptoms might have been noticeable, but the analyst could not interpret them. He could perhaps diagnose the fault mode, but not the severity. If the fault has been misdiagnosed once and was not verified, it could be misinterpreted again.

Another problem is introduced, when two or more fault modes appear simultaneously. Some of the symptoms might be a result of a root cause, such as imbalance or misalignment, while other symptoms might be related to the resulting fault modes, such as a bearing defect.

A classifier relies on the weight vectors that have been defined using data samples and syndromes. In order for a classifier to give a plain text diagnosis and severity as an output, it needs to be calibrated or labelled. This means that all classes, which were trained using data samples from a known condition, should be made identifiable. Typically the training samples used to train a single class would have a same interpretation. If not, the symptom extraction might have been imperfect. In other words, the symptom extraction chosen could not differentiate between two (or more) fault modes and should be improved. If this is not possible, the interpretation of a new sample falling into this particular class, would mean that in some probability the new sample represents any or combination of all possible fault modes.

The probability of a failure is very difficult to be defined as an absolute value. Fuzzy logic may be here useful. Let's assume that the classifier has been labelled and verified correctly so that each class has been given a correct interpretation. This should include the fault mode or modes, if there are many in a single class, and the severity. The description of a fault mode can in many cases be deterministic, but for severity fuzzy groups may have to be used. The labelling should be verified using strong evidence, such as confirming the fault mode and severity by other methods.

When interpreting a new sample, a classifier looks for a best matching class, which obviously will always be found. When comparing the new sample with the samples used to train the class, an assessment can be made on the membership within the class. If the new sample falls close to the geometric centre of the training samples, the membership is high. If on the other hand, the sample is further away from the centre than any of the training samples, the membership is low. Taken that the classifier has been accurately labelled, the confidence of diagnosis would then be high or low consequentially.

5.8. Generalization

The prediction of the condition of an individual machine is difficult, because the reference data are generally deficient. The amount of existing measured data is scant especially at the initial phase and such data are not available in all operational states of the machine. Therefore, the detection of anomalies cannot be based on the historical data, but general information is utilized instead. Consequently, the anomaly detection is uncertain and inaccurate. [Lumme, 2004]

The absence of empiric data is a special disadvantage in the diagnostics of faults. In practice, fault diagnosis is based on known symptom rules, which have been widely published. In many cases, the rules are of a general nature and not based on the descriptor or symptom values obtained from the machine in question. Finnish patent [Lumme, FI102857] discloses a method, whereby a system can be made to learn from measurement results. However, the problem here is that in order to perform a fault diagnosis, an individual machine must first experience all the faults that the system is expected to identify. This is not possible in practice, which is one of the reasons, why no viable remote diagnostics systems have been developed. Even if some kind of systems of this type do exist, they are only capable of solving simple diagnosing tasks based on using a single or a few symptoms, but they are unable to handle syndromes.

A European patent [Lumme, EP1292812] discloses a method to eliminate the above mentioned disadvantages. Measurements are collected on a maximal number of preferably, but not necessarily, identical machines or machines of substantially the same type to obtain descriptors of the operation and condition of machines. As a result, any fault mode detected on a single object can be generalized to be applied in the diagnostics of the similar fault mode in any other object of a same type.

The essential point about the usage and functionality of the patented method is that the database should be as large as possible and contain characteristic vectors descriptive of different operational states of machines in question in as large an area of application as possible. The method has proven effective in the anomaly and fault detection in a gearbox application with a large number of descriptors, which will be discussed in detail later in chapter 6 [Lumme, 2011].

Figure 19 presents a classifier map that has been created based on data collected from several similar objects. The background colour is used to show the class labels, i.e. fault modes. A case is initiated, when an anomaly is first detected. It consists of one or more events during which the data records were collected. Each event includes all information related to a specific anomaly and fault progress. As a fault propagates through its various modes, it shows as changing vibration patterns, which are then used to train numerous classes.

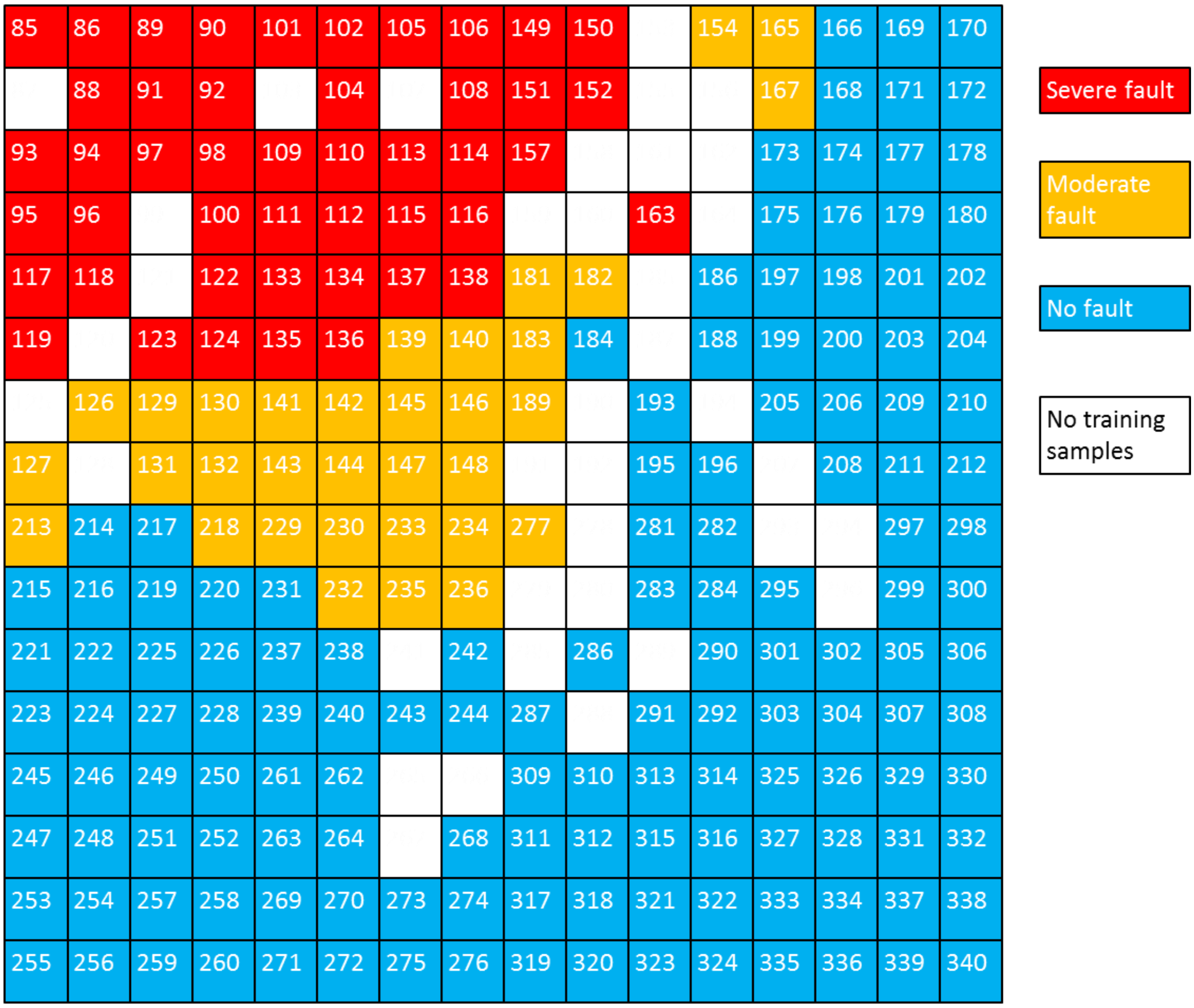

Figure 19: Generalized map shows several severe fault modes in red in the top left corner and moderate fault modes in the middle. Several classes are used to represent various fault types and modes.

5.9. Summary of the method

We have now addressed all the stages of the continuous learning. Figure 20 summarizes the training and retraining processes. The procedure is slightly different for retraining, where the class labels and weights shall be memorized before the training process and retraced after the process. The system shall allow the user to label the classes at any time, except between the memorization and retracing processes. Class labelling may happen typically after any retraining process, but can wait until the diagnosis and prognosis have been confirmed.

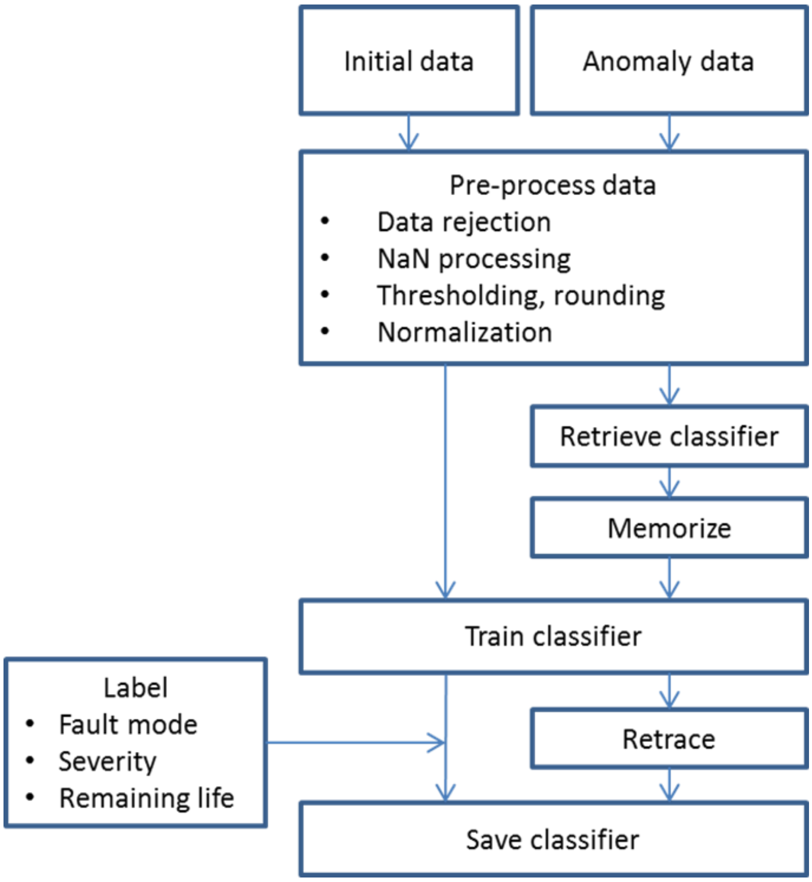

Figure 20: The training process first starts with the acquisition of initial data, which then needs to be pre-processed. The processed data is used to produce a classifier that consists of a number of classes descriptive to the characteristics of the data. An effort should be made to label the classes using human expertise. Whenever an anomaly is observed, the related data should be acquired and processed as with the initial training. The re-training process is basically the same, but the class weights and labels should be memorized before re-arranging the classes and the respective classes should be retraced and saved.

The prediction procedure is summarized in Figure 21. Data samples can be tested one by one or in a batch. After the data pre-processing the proper classifier shall be retrieved from the database. There might be several classifiers available for even a single object. It is recommended that descriptors that are related to different kinds of characteristics or different timestamps use their own classifiers. For instance a classifier with vibration, temperature and oil descriptors mixed might fail, because the parameters may have been collected at different spans.

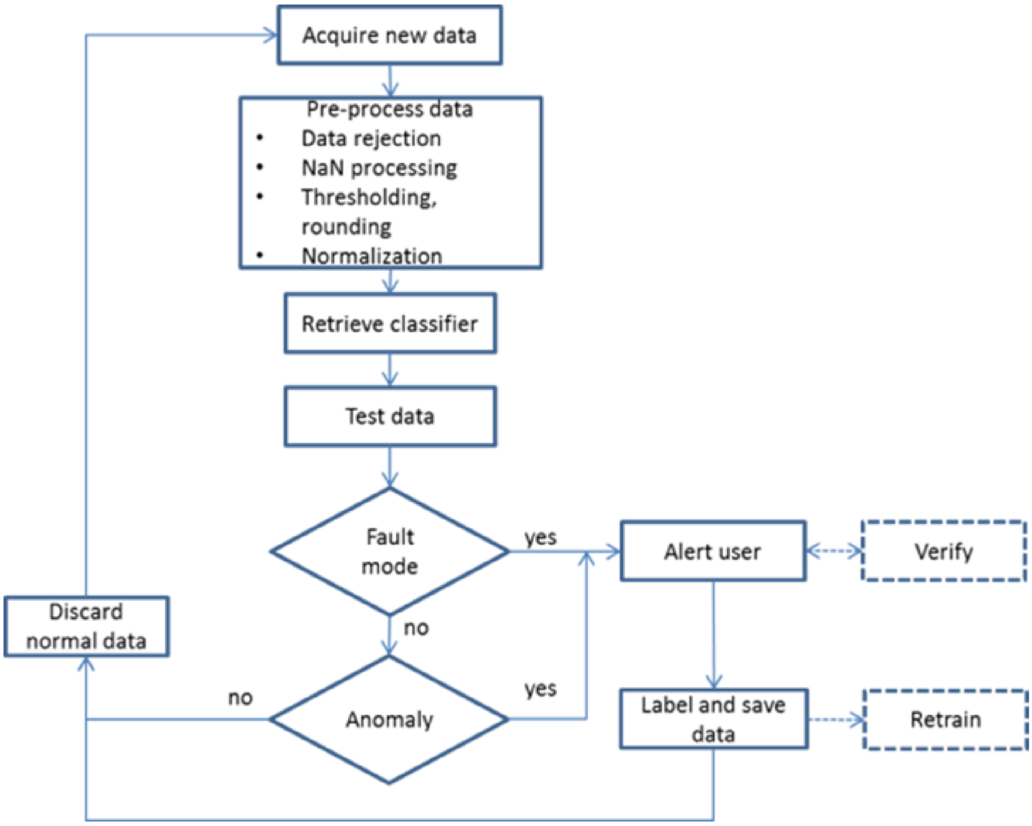

Figure 21: Prediction process is a simple sequence to test any new data. After pre-processing, new data is compared against all known classes. A best matching class will always be found and the user should be alerted, if the class has a label of a fault mode. Secondarily, it shall be tested, if the new data has anomaly characteristics, in which case the user should also be alerted. In both cases the user should exercise his expertise and either verify or label the data for retraining purposes.

After finding the best matching class it is important to test first, if its label refers to a fault and only after that if the data is an anomaly. In fact it might not even be necessary to test separately, if a data sample classified as a fault also is an anomaly. Each fault occurrence will always have variations mainly because of various degrees of fault severity. All data related to fault modes is valuable and will increase the confidence level of the class label. However, before proceeding with the retraining, both the anomalies and fault detections should be verified by human expertise. This may include removal of outliers.

All data samples that were classified as normal can be discarded. Typically there is already enough training data for classes labelled as normal mode and there is no further use for such data. In fact, this brings a major benefit in optimizing the data to be saved. Many conventional condition monitoring systems keep collecting and trending data that stays stable and has only a little or no use in the prediction of machine condition. [Lumme, 2000]

6. EXPERIMENTS WITH MULTIVARIATE DATA

6.1. Introduction

This chapter presents results of the experiments with multivariate data, which has been collected from several wind turbine gearboxes in many different wind turbine parks. The purpose of this study is to demonstrate the ability of a classifier to provide satisfactory predictions in a very demanding application. The experiments are designed to comprise of all classification processes presented earlier in this thesis. Also the various difficulties that could affect the confidence of prediction will be discussed. The typical planetary gearbox design is shown in Figure 22.

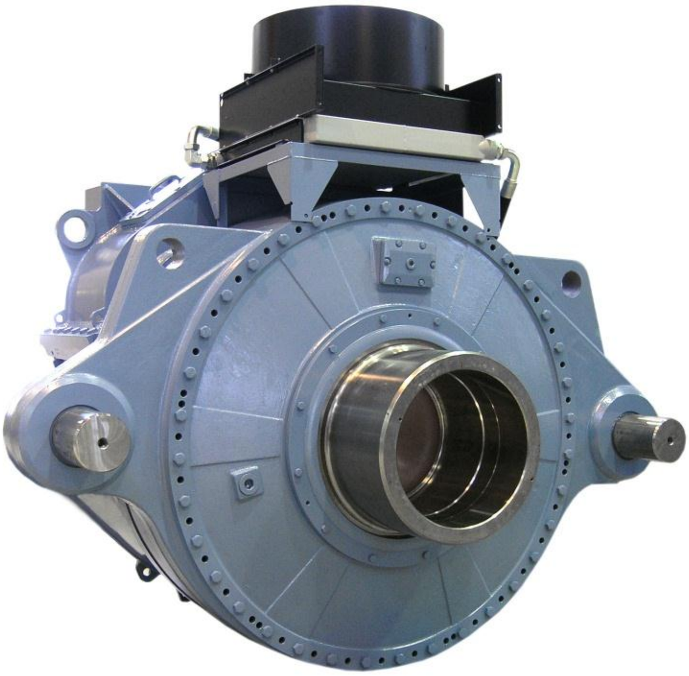

Figure 22: Typical planetary gearbox design (Moventas MM82) used in wind turbines.

Moventas has established a remote monitoring center in order to remotely monitor and report on the condition of wind turbine gear transmission components. The system consists of various sensors, a central unit and user interface with analysis software. The sensors acquire data from the gear transmission. Data is collected by the central unit and transmitted to a server. The analysis software and user interface offer tools for the analyst to manage the condition of monitored gear transmissions. The general principle of the Condition Management System (CMaS) in illustrated in Figure 23. [Elfström, 2011]

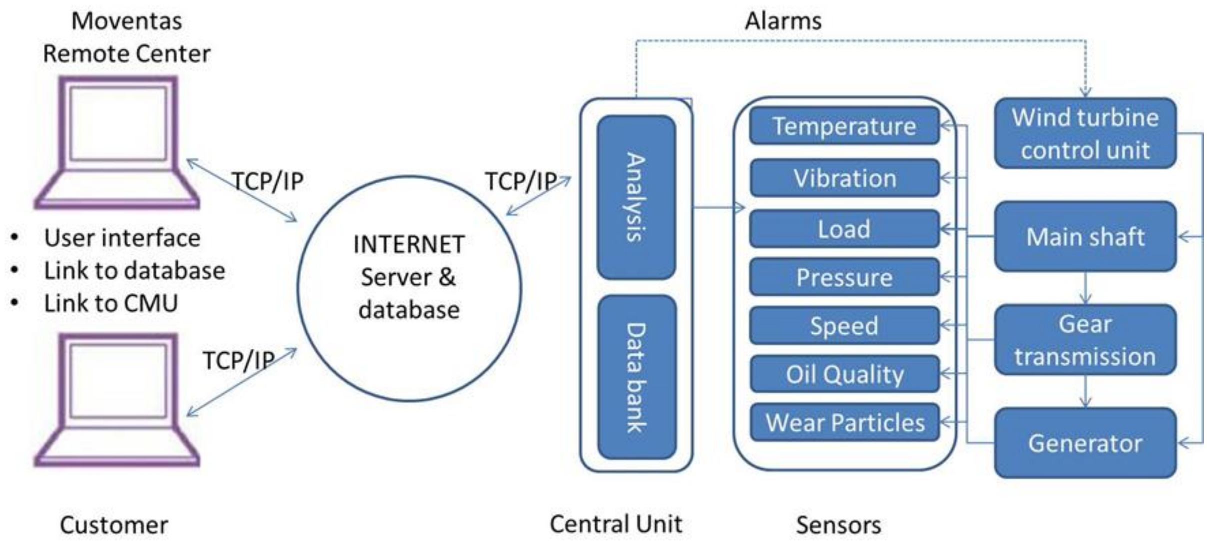

Figure 23: Condition Management System (CMaS) used to collect and analyse the data for the experiments

The purpose of the system is to predict the actual maintenance needs to avoid unexpected maintenance actions. Normally the costs of an unscheduled shutdown are thousands of euros per day. For instance, a minor fault in the cabin may progress to a failure requiring maintenance in a workshop, which often involves taking down the gearbox, generator, main shaft with bearings and propeller. In such a case, the repair costs will be eminent because of the expensive hire of the lifting equipment. The monitoring enables the failure detection in time, which allows the shut-down to be planned and organized within a few days.

The corner stones in the system design were:

- the measurement of descriptor predicting the condition of a machine
- the system modifiability using smart sensors
- the remote update of the CPU
- the user interface via Internet browser
- the ability to measure large amounts of objects cost efficiently
- concentration on effective remote monitoring, not in problem solving

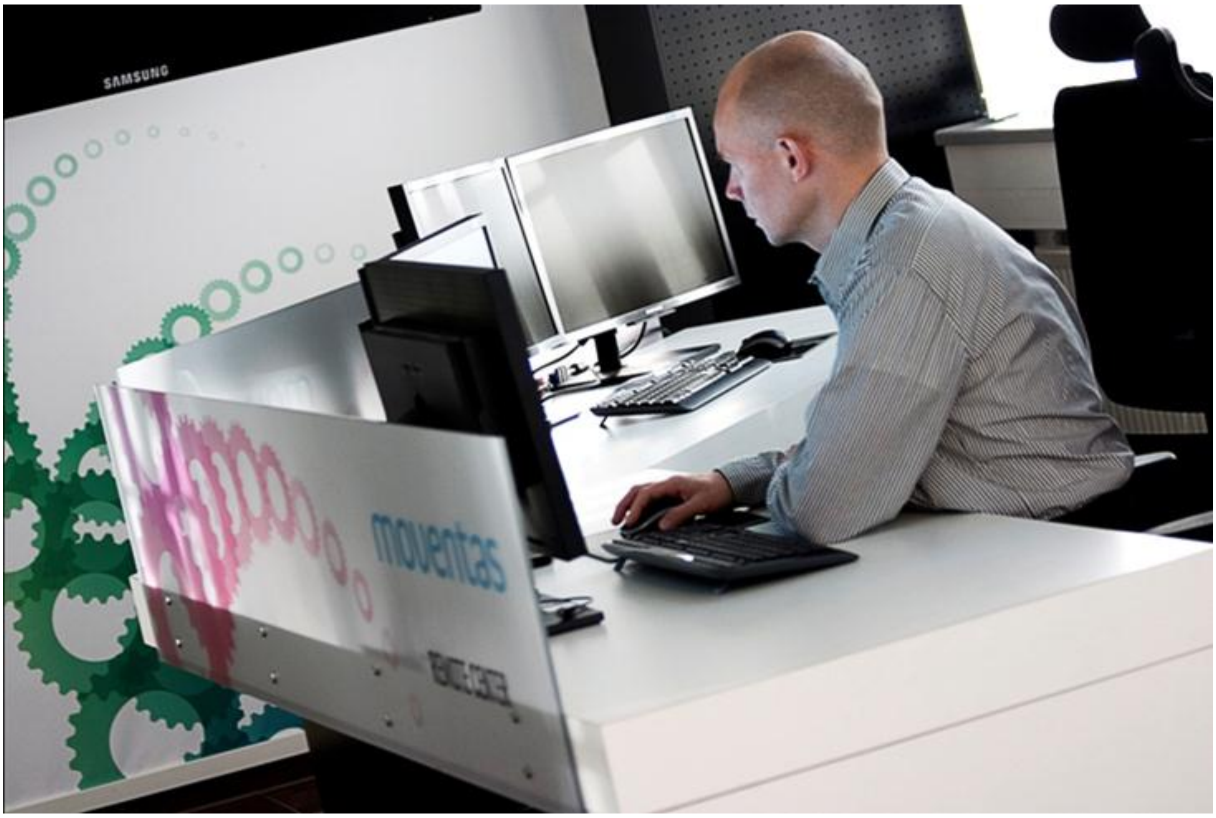

Figure 24: Remote wind turbine gearbox monitoring centre in Jyväskylä, Finland.

The core of the system is a smart sensor (Figure 25), which was a result of Moventas internal development. The sensor carries out basic analysis, such as transform of time domain to frequency domain and envelope detection. In this way, there is no need to transfer weak signals in a noisy environment. Also, the quality and amount of data can be optimized. The sensor can be programmed remotely. As no signal processing is needed in the control unit, it is very affordable.

Figure 25: Smart sensor (size ø 54 mm, height 44 mm) used to acquire vibration data for the experiments. The digital sensor acquires the vibration data from the gearbox, performs a spectrum analysis and feature extraction and makes the data available to be collected by the on-line monitoring system (CMaS).

The monitoring of a modern wind turbine transmission system is in many respects much more demanding that monitoring of conventional transmissions. The biggest challenge is the varying operational conditions. The turbine load and speed are changing constantly. The speed of the shaft between the generator and gearbox can in fact drop in a couple of seconds from 2000 rpm to 200 rpm.

One wind turbine park may consist of tens or even hundreds of wind turbines. Most of the measurements and analyses can be automated, in which case one analyst can monitor about 300 turbines. By enhancing the data acquisition and analysis methods Moventas aims to a system, where one person can cover more than 500 machines.

Moventas' CMaS system produces among others 126 vibration descriptors every 90 minutes. Due to the large amount of descriptors, it is very difficult for an analyst to handle the data volume in order to detect anomalies and faults. At the minimum, it would be desirable to have a system that could detect, if the monitored gearbox is in a normal condition. As a secondary objective, the system should be able to utilize the data and experience achieved from all fault modes and their progression on any of the monitored gearboxes. This data and experience should be transformed to general knowledge or even intelligence to support and automate future prediction of similar failure patterns.

Typical descriptors extracted from a vibration signal by the CMaS system are:

- vibration amplitude at rotational frequencies and harmonics
- vibration amplitude at roller bearing forcing frequencies and harmonics
- vibration amplitude at gear mesh frequencies and harmonics

A four layer classifier with 64 classes was used in this experiment for demonstration purposes. Actually a five layer classifier with 256 classes would certainly have been more sensitive, but it is more difficult to visualize on paper.

This study does not take a position in the selection and justifying descriptors and symptoms. It is, however, probable that due to the great amount of descriptors some of them are highly dependent. The main objective is to test and demonstrate the multivariate classifier in an utmost demanding environment with a significant amount of error sources.

6.2. Data pre-processing

As there are numerous descriptors for each measurement event, their values may vary significantly depending on the wind, load, speed and fault modes. Occasionally, single descriptor values or even the whole data set might be missing. The data set might also be erroneous partly or as a whole. Some of the descriptor values are extremely low compared with the others. Some vary significantly. At the moment, all descriptors are expressed in same units (mm/s), but in future this might change. In order to cover all these aspects several data pre-processing techniques were used.

The rotation speed of the gearbox varied between 0 to 1800 rpm. During the preliminary analysis, it became clear that vibration data at low speeds is not really descriptive to the condition of the gearbox or its components. When low speed data was originally included in the training data, the classifier was distorted and the prediction confidence reduced. To avoid this, all data collected below 100 rpm was discarded.

In a machine monitoring application, the descriptors should change only gradually in small increments, when the measurement interval is short. During the initial training, the maximum distance is saved for all classes. The maximum distance gives the longest distance of the training samples from the weight center. If the maximum distance is exceptionally long compared with the other training samples or other classes, this is caused by an outlier. Also, the number of training samples is low for a class, where one of samples is an outlier. An outlier left in the training data will cause an error in the membership calculation. During prediction, outliers can be detected by comparing data samples with the previous and next samples.

The variation in the low amplitude descriptor values was found to be mainly random and not a result of changes in the gearbox operation or condition. Because of the normalization process, which in this case was the standard score method, random meaningless variance at low magnitudes resulted in high symptom values, which again gave excessive weight on these descriptors in the training of a classifier. Therefore, a certain threshold value, in this case 0.1 mm/s, was applied. Any descriptor value falling under the threshold value was set to zero. This method actually adds diversity in the symptoms. On the other hand, it may turn out that none of the values for a specific descriptor exceeds the threshold value. This would prevent the use of standard score normalization, where standard deviation is used as a divider. In any case, for the samples with a mean descriptor value not exceeding the threshold value, division by the standard deviation is not used. As a result, descriptors with small mean values are downgraded before the training process. Resetting low magnitude values to the threshold value could also be useful, but this was not tested during this work.

Due to various reasons data was not always available for a certain descriptor or any of the descriptors in a data sample. It is not advisable to replace missing values by zero, because it has a definitive value. This would falsify the syndrome and therefore result in a poor confidence of the classifier. The problem was resolved by replacing all missing values with Not a Number (NaN), which is a numerical data type representing an undefined value. Most computer processors and software compilers accept NaN as a floating point number, but they may give different results in operations, which brings a new problem. A missing descriptor value should really be replaced with the average of the same descriptor values within the similar data samples, i.e. data samples classified in the same class. The classification result of samples is not known before the training of the classifier and therefore it was concluded that adequate confidence is achieved, if the missing value were replaced with the mean value calculated from the total training population. In practice, this can be accomplished by replacing the NaN values during the normalization with zeros. In case all descriptor values for a data sample were missing, the data has no importance and should be discarded. A decision should be taken on, how many descriptors are really needed to confidently represent the current state. This was not tested, but roughly at least half of the descriptor values should be present.

Before training of a classifier, the descriptor values were normalized using the standard score method. As discussed earlier, subtracting the mean values (normal values) from the actual values transforms the descriptors to symptoms. Division of the symptom values with the standard deviation values makes the symptoms dimensionless and commensurable. The data population used to calculate the mean and standard deviation values was changed throughout the initial training, testing and retraining processes as explained later.

Finally, it is important to understand that the descriptors should have been selected so that a minor fault progression would cause a detectable change (i.e. a symptom) in the descriptor value. If this was not the case, there is nothing much to be done to improve the confidence of prediction. Consequently, a small change in the symptom value is insignificant and probably only caused by random variation. For this reason, all symptom values were rounded to a precision of two decimals. The purpose of this operation was to emphasize the actual diversity in the data values.

6.3. Initial training

The data for the initial training was collected from a single wind turbine gearbox "GB01" in United Kingdom between September 1st, 2008 and June 1st, 2009. During the data pre-processing, about 300 low speed measurements and two outliers were discarded, after which 3534 data samples remained. It was known in advance that there were no developing faults during year 2008. Therefore, 1776 data samples from this period were selected for initial training. To accelerate the adaption, the data population from the total period until June was used for calculating the means (offsets) and standard deviations (scales) for the normalization. Ultimately, in a practical application it would be preferable to use as large a data population as possible.

Figure 26 illustrates the classifier map after the initial training. Because all classes represent the normal state of the gearbox, it is not worthwhile drawing any conclusions on the classes. The numbers on the top left corners give the index of the class and the number of the training samples in the brackets. The scattering of the data over the map is caused by the varying operating conditions, such as wind speed and direction, rotating speed, etc. It appears that almost all classes have been populated to show only normal modes and one might conclude that there is no space left for the eventual fault modes in the future. We will, however, see that the classifier adapts to changes in data.

21 33	22 35	25 104	26 27	37 43	38 11	41 33	
23 51	24 43	27 43	28 9	39 21	40 55		44 119
29 43	30 14		34 25	45 15		49 60	50 4
	32 9	35 13	36 21		48 39	51 15	52 62
53 14	54 20	57 19	58 21	69 3	70 3	73 73	74 60
55 21	56 20	59 9	60 8		72 22	75 50	76 55
61 22	62 17	65 21	66 12	77 21			82 178
63 25	64 22	67 22	68 24	79 19	80 42		

Figure 26: Classifier map after initial training shows the index (top) and number of training samples (bottom) in each class.

6.4. Prediction

The second set of data samples starting from January 1st, 2009 was used for prediction. The same offsets and scales were used in the normalization. After initial training, the classifier cannot be used for fault detection, but for anomaly detection only. The class membership was determined for each new data sample. The anomalies will show as low membership of the data samples within the classes. For demonstration purposes, the membership is now presented in reverse using an anomaly index, which is determined by dividing the distance of the data sample from the weight center by the maximum distance of all datasets used to train the class. Therefore, low membership means high anomaly index and vice versa. This yields a conventional presentation, where undesired results are seen as increased magnitudes on the y-axis. See Figure 27, where the y-axis is logarithmic to emphasize the changes. Note that the x-axis is not linear, because of the discarded data samples.

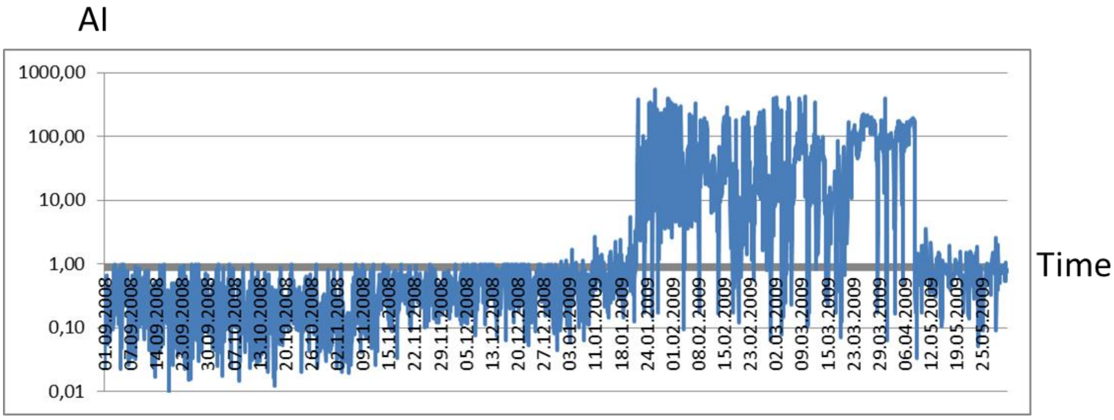

Figure 27: The plot showing anomaly index (AI) versus time reveals major changes at the end of January 2009.

The high membership of the training samples can be seen as low anomaly indexes at the left side of the plot. For each trained class, there will always be one data sample, where the anomaly index value reaches 1.0. The figure shows a few short duration, minor anomalies in the beginning of the year 2009. A major change occurs around January 20th, where after very high anomaly values can be detected. It can be concluded that the data samples collected after this date are not members of any classes and present a totally new syndromes. The data reveals that the symptom values have increased significantly compared with the weight values of the best matching classes. Figure 28 gives an example of the symptom values for data sample with an anomaly index 556 collected on Jan 27th, 2009 at 21:00. The respective descriptor values are shown in Figure 29.

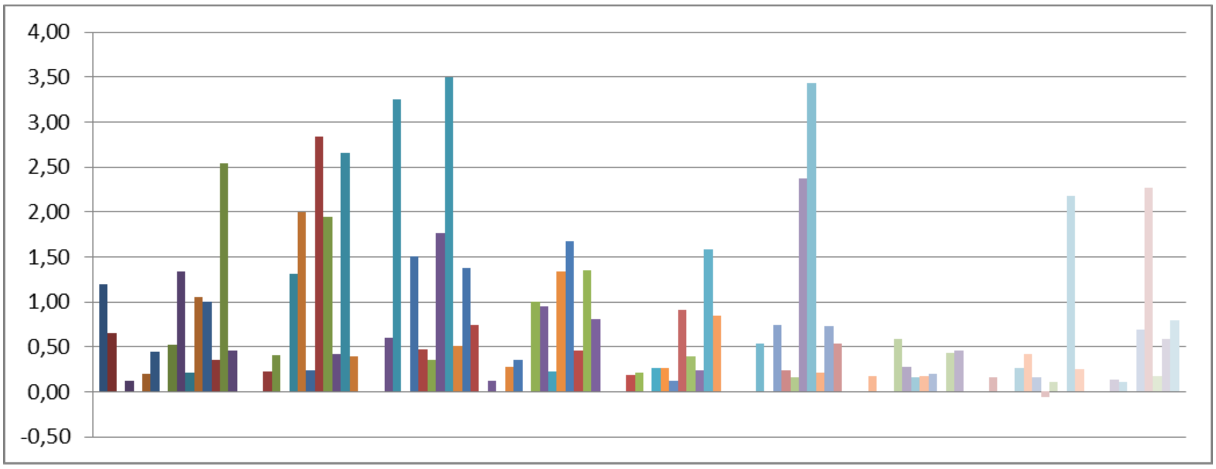

Figure 28: Symptom values indicate the changes in descriptor values from average values. The distribution of symptom values helps to understand the nature of the fault mode.

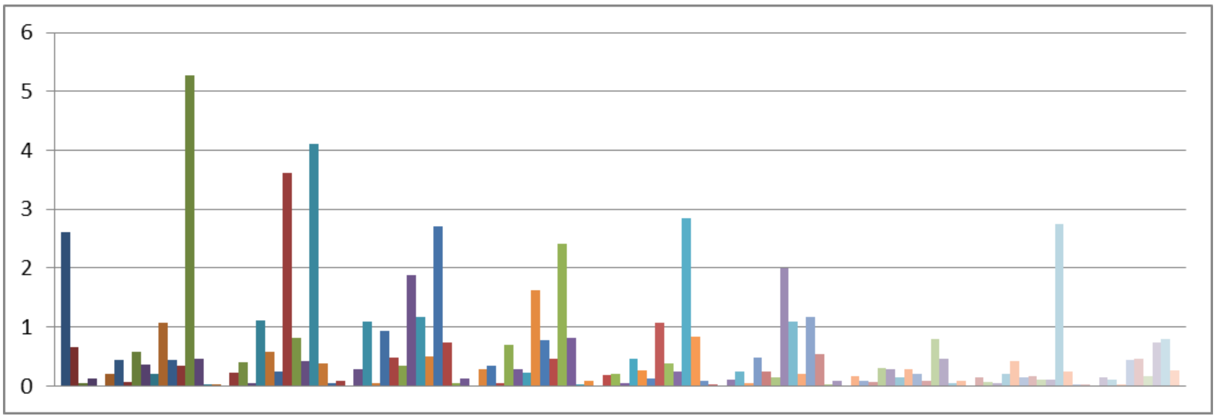

Figure 29: Descriptor values show the absolute amplitudes of the 126 extracted values considered effective to define the gear condition.

By analysing the raw data, it was concluded that the anomaly was caused by a bearing fault on the high speed shaft of the gearbox. The bearing was replaced in April 2009, where after the anomaly index dropped, but not always under 1.0. This indicates that there are still minor anomalies after the bearing replacement. Altogether, there were more than 1089 data samples in year 2009 exceeding the selected anomaly limit of 1.3.

6.5. First re-training

As the number of anomalies was relatively high, it was decided to include all data samples from the year 2009 to the training data. An attempt will be made to identify the classes representing the fault modes on the new classifier map. The re-trained classifier map is shown in Figure 30.

21 24	22 22	25 18	26 65		38 81	41 78	42 53
23 37	24 6	27 37	28 19			43 47	44 98
29 26	30 44	33 21	34 4	45 102	46 196	49 2	50 118
31 26	32 13	35 8	36 14	47 120	48 93	51 13	52 100
53 48	54 43	57 38		69 220		73 92	74 108
55 36	56 22	59 46	60 52	71 2	72 151	75 90	76 28
61 33	62 21	65 16	66 60	77 35	78 86	81 29	82 153
63 22	64 32	67 44	68 54	79 47	80 108	83 258	84 76

Figure 30: Classifier map after first retraining shows changes in the organisation of the classes as new data has been appended in the training set. Classes 21 to 35 are solely populated with the anomaly data.

Due to the initialization process in the training algorithm, the classes at the top left corner of the classifier map tend to represent symptoms with high values and the classes at the opposite corner low values. Figure 31 shows the distribution of symptom weights in class 21 and Figure 32 the respective symptoms in class 84. Because of the standard score normalisation, where the mean value of the population is subtracted from the raw descriptor value, the symptom values are negative in the bottom left corner.

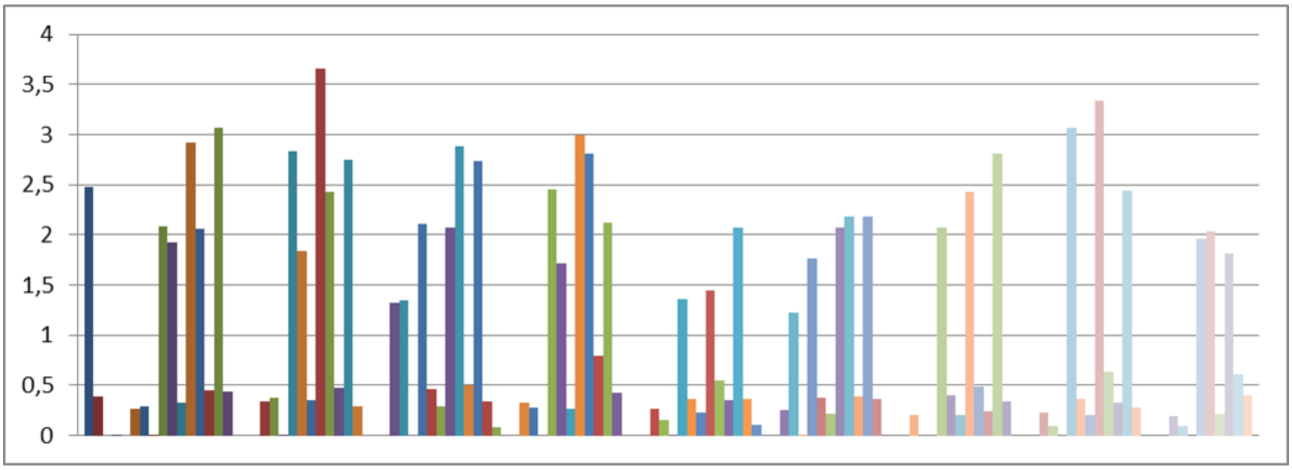

Figure 31: High symptom values are typical in class 21 (top left corner of the map).

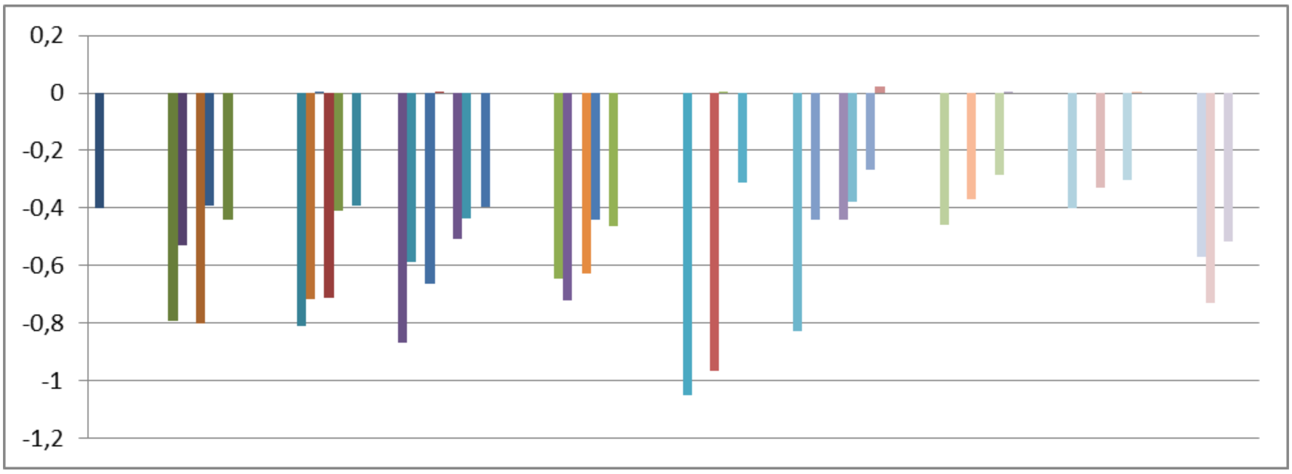

Figure 32: Low symptom values are typical in class 84 (bottom right corner of the map)

By studying the classification of the anomaly data on the re-trained classifier, it can be concluded that this data has been used to train classes in the top left corner without any data samples from the initial data set. This is self-explanatory, because of the great diversity between the initial and anomaly data. The classes from 21 to 35 are populated by the anomaly samples with high anomaly indexes only. The classes from 36 to 52 represent samples collected during a normal condition. The classes from 53 to 68 are also populated by anomaly samples only and the classes from 69 to 84 by samples with high class membership.

All classes, except for 36, in the left half of the map represent the bearing fault or faults with its various stages of progression. Figure 33 shows the Euclidean distances for all samples falling in classes 21 to 24. The distance is here used instead of membership or anomaly index to maintain the relationships. A progressive trend can be observed in the plot indicating that the fault mode is becoming more severe. Figure 34 shows the distances for samples in classes 53 to 72. Note that the dates are partly overlapping suggesting that the state is fluctuating between two modes, which is probably caused by changes in the wind conditions.

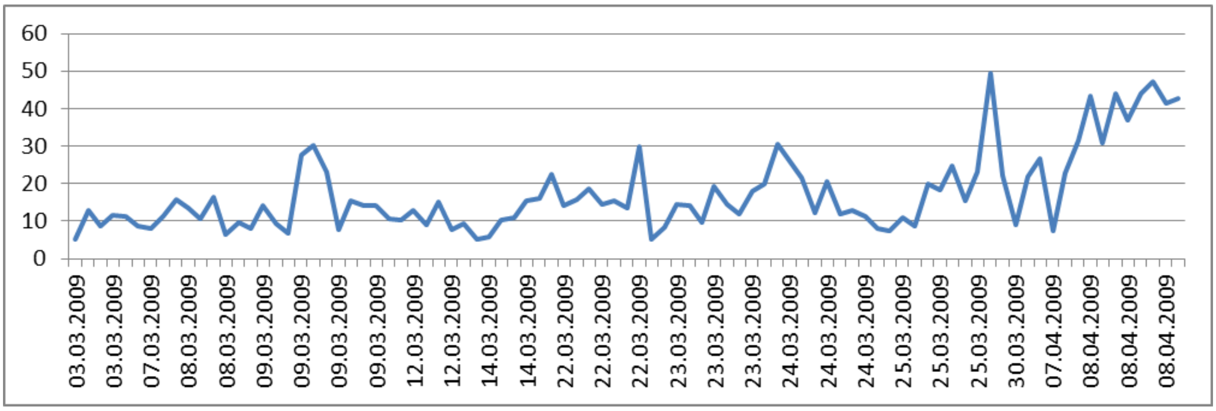

Figure 33: Euclidean distances of data samples from respective weight centres in classes 21 to 24 indicate a fault mode becoming more severe.

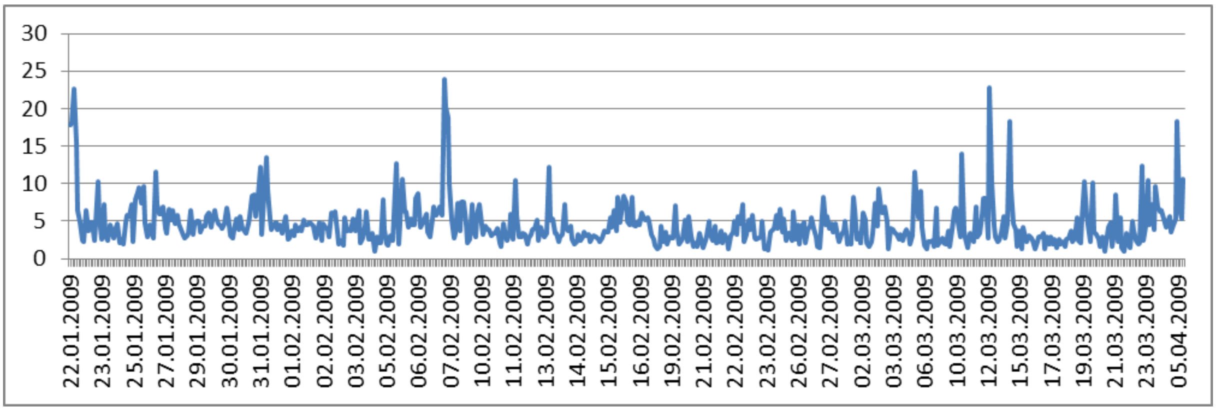

Figure 34: Euclidean distances of data samples from respective weight centres in classes 53 to 72 shows overlapping dates with the previous plot.

The classification principle discussed in this thesis makes no attempt to label the classes automatically after the first re-training. There is no mechanism for the classifier to conclude, which classes should be labelled as representatives of fault modes. Even for a human expert it is often difficult to determine the exact beginning of the fault progression, because it is often a fuzzy perception. The best solution is to revert to the original data and aim to determine, which classes represent normal and early fault modes. For instance, a data sample collected on Jan 25th, 2009 was classified in class 53 and another data sample collected on March 9th, 2009 was classified in class 21. The descriptor values of these data are shown in Figure 35 and Figure 36 respectively. We can clearly observe that the data vectors are almost equal, but the magnitudes are higher in the later sample. By continuing this exercise, we will finally get an impression on, where to set the limits between the classes representing normal and early fault modes.

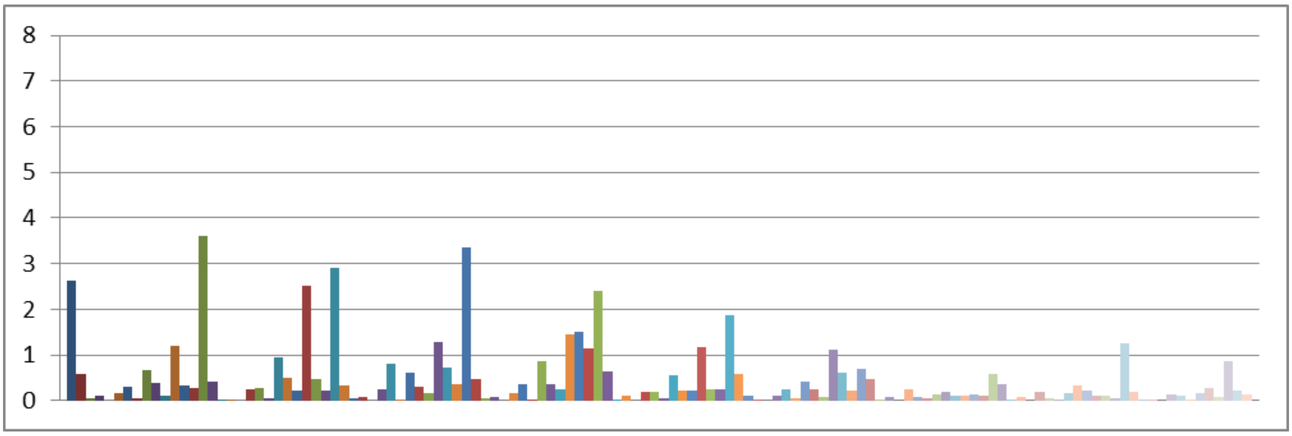

Figure 35: Descriptors of a data sample in class 53 have moderate values suggesting a normal or early fault mode.

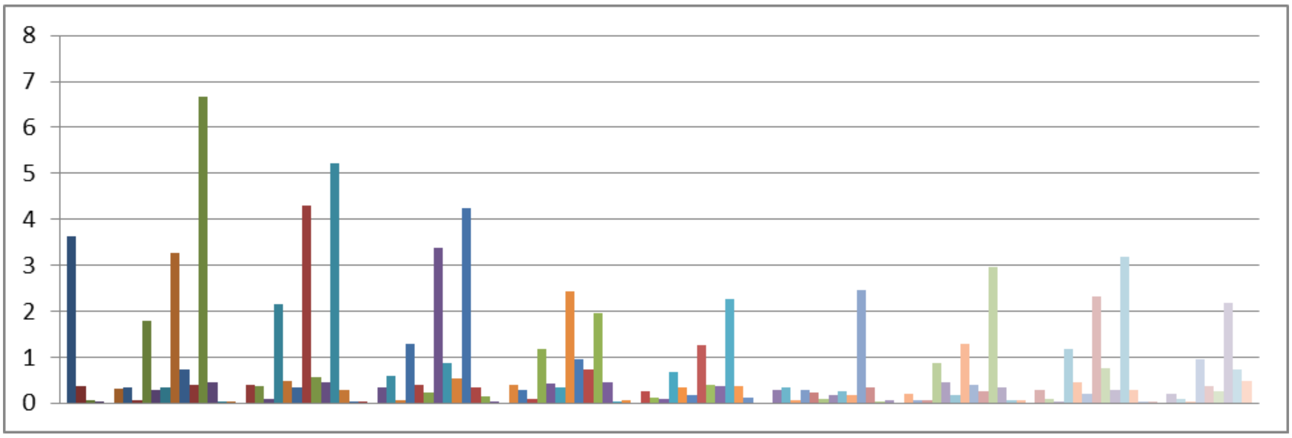

Figure 36: Descriptors of a data sample in class 21 have high values compared with those in class 53 suggesting a severe fault mode.

We can also use a trajectory presentation to illustrate, how the data classification has proceeded. Figure 37 shows the historical view to the classification. Note that only a few data samples before and after transitions have been marked to avoid confusion. Before Jan 20th, 2009 all data was classified in the classes on the right side of the map. Afterwards, all data was classified in classes 53 to 66 excluding classes 63 and 64. Finally after March 9th, 2009 most data samples were classified in the top left classes.

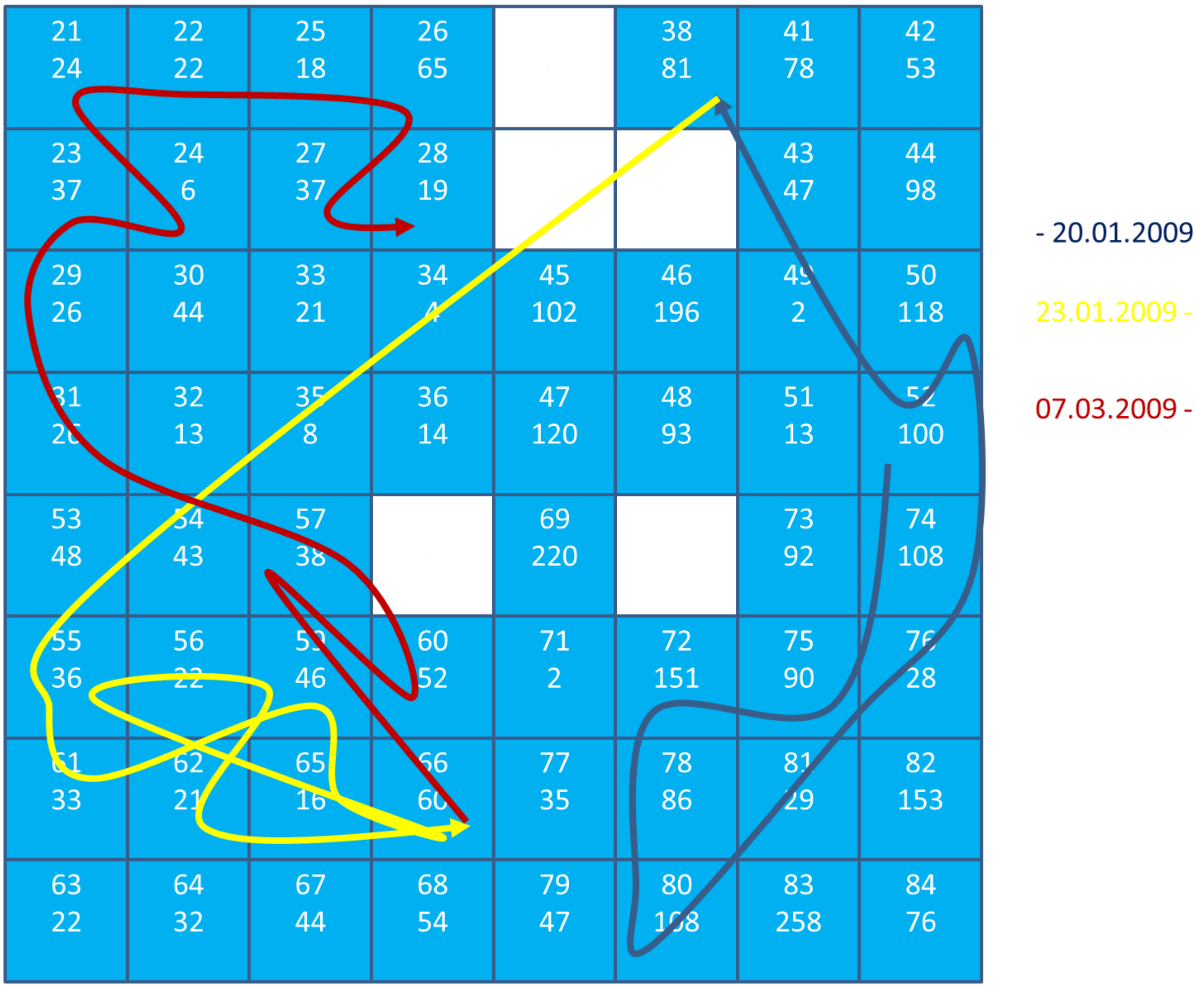

Figure 37: Trajectory view shows how the progress of the fault mode can be visualized on the classifier map. The plot illustrates three different stages: no fault (blue), moderate fault (yellow) and severe fault (red). The fluctuation from class to class within a short interval is caused by changing wind conditions.

The figure shows the classes hit by yellow and red curves that present the fault modes and should be labelled accordingly, for instance as a bearing fault. An attempt should be made to define also the severity of a fault mode for each of these classes. Classes 30, 33, 34 and 35 were also hit by data samples during the same period. However, because of their lower descriptor values and earlier appearance in the sequence, they were left to represent normal states. The decision on the classes to be labelled affects directly on the under- or over-diagnosis in future predictions.

6.6. Fault detection and progression

The classifier created above is now able to reliable predict the label of any data sample in the training data set. This can be easily tested and results in all samples being classified accordingly. There might be a minor deviation so that some of the samples fall in the adjacent class. Sometimes the adjacent classes are very close in hyperspace and such a deviation is tolerable.

A real test can be conducted by using a totally new data set for prediction. Measurement data between Aug 1st and Dec 31st, 2009 from a second identical turbine gearbox “GB02” in the same wind turbine park was selected for prediction. After data pre-processing 2270 data samples were available. It is not necessary to discard outliers in advance, because they will be detected during the prediction. The same offset and scale factors created during the re-training with the total population from gearbox “GB01” was used for normalization.

Figure 38 illustrates the distances to weight centres for those data samples that were classified to any of the labelled classes, i.e. classes representing fault modes. It suggests that starting from Nov 19th, 2009 the gearbox is experiencing the same fault mode as the gearbox “GB01” earlier the same year. In order to confirm this, the anomaly index should be analysed, but the user should be alerted already at this time.

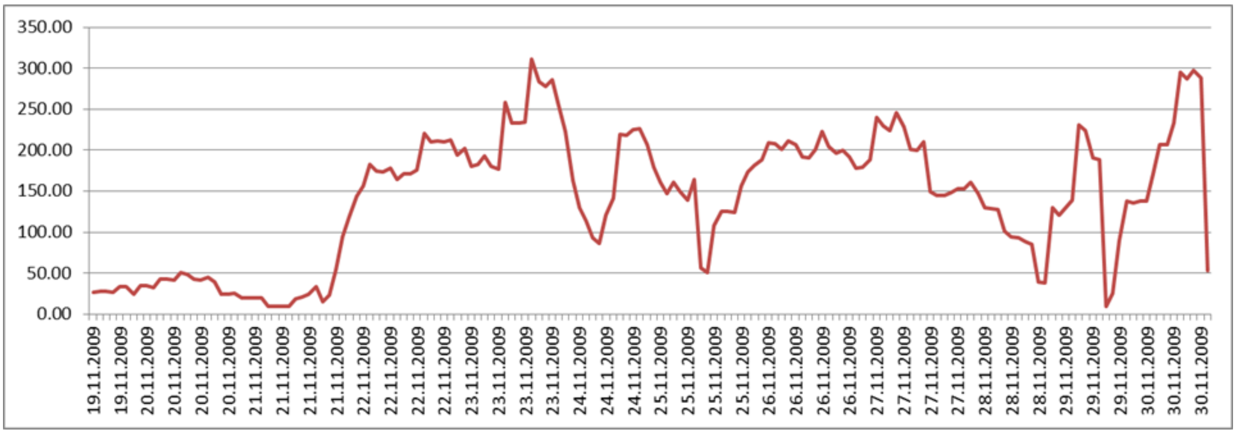

Figure 38: Distances of data samples from the weight centres of the best matching class in labelled (fault mode related) classes are being elevated as the fault progresses.

Figure 39 shows the anomaly index for the whole time period and reveals that there are several anomalies in the data set. Some of these coincide with the date range of Figure 38 indicating that the fault mode is not precisely similar with the one on the other gearbox. The anomalies in early August can be considered random and should not cause alerts. However, the anomalies starting in mid-September are longer in duration and have significant index values.

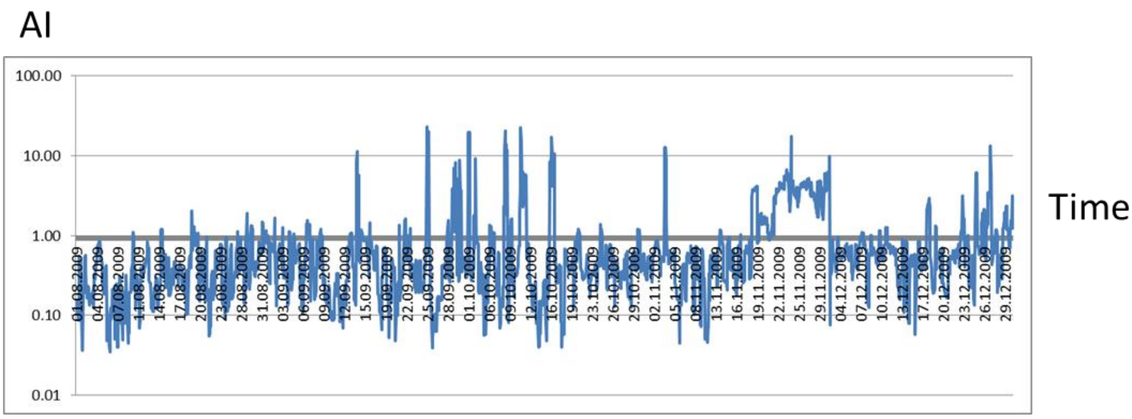

Figure 39: Anomaly index trend for the whole data set show random anomalies in August, but more continuous anomalies in mid-September.

Let us study the trajectory plot again in Figure 40, which shows the travel paths just before transitions. We can see that the first anomalies happen in mid-September in classes 38, 41, 43 and 51. When compared with the previous trajectory in Figure 37 with data from the gearbox "GB01", it can be seen that these classes are linked to the events just before the transition into the fault mode. Apparently, we are gaining more information on the early warning and the fault progress. In order to use this information, we need to re-train the classifier. Before that it is necessary to memorize the de-normalized weight factors of all labelled classes (21 to 29, 31, 32, 55 to 62, 65 and 66). The class numbers are used as labels in this case for an illustrative purpose, but in a real application fault mode and severity would be more appropriate. De-normalization, which is necessary, because the mean and standard deviation will change, is done by multiplying each weight by the standard deviation (scale factor) and adding the mean (offset factor).

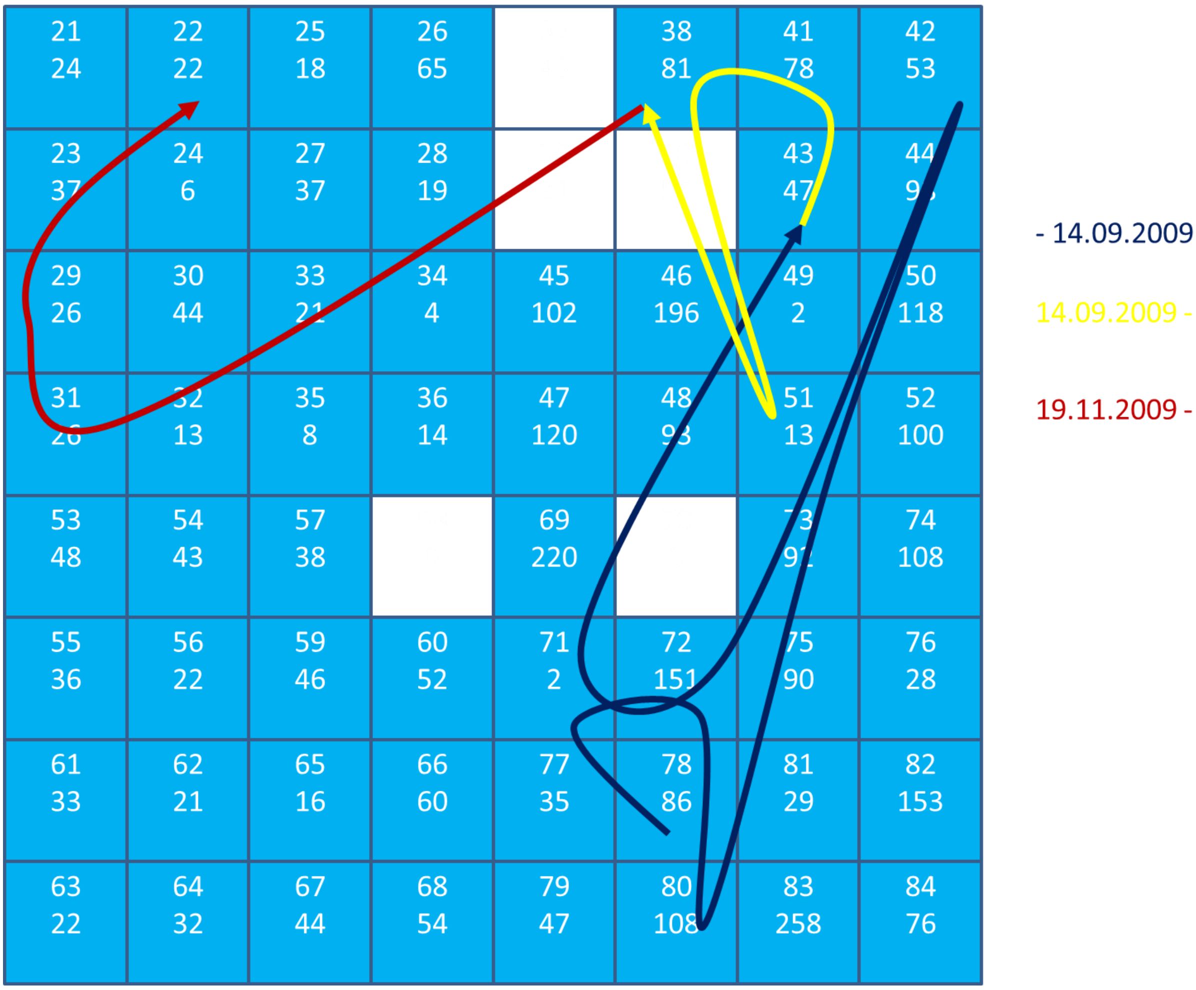

Figure 40: Trajectory view showing fault progression as transitions on the map. The upper number in the class gives the class index and the lower number the amount of data samples used to train the class.

We already have a reasonable amount of training data for the normal modes and therefore append only the anomalies to the data bank consisting now of 6136 data samples. All data will be normalized using the original and appended data population. The retrained classifier map is given in Figure 41. We can now see the classifier's ability to adapt to the new data.

21 88	22 64	25 32	26 8	37 60	38 78		42 242
23 7	24 16	27 32	28 2	39 22	40 17	43 160	44 120
29 83 25, 26	30 32 28	33 12	34 200	45 99		49 104	50 111
31 35 21, 22	32 49 24, 27			47 129	48 322	51 37	52 182
53 80 23, 29	54 49	57 49 61, 62	58 61	69 2	70 23	73 129	74 120
55 44	56 40 55	59 33	60 107 66	71 94	72 193	75 153	76 152
61 72 31, 32	62 33 56	65 66 60, 65	66 47	77 254	78 183	81 137	82 53
63 53	64 53 57	67 42 59	68 6	79 294	80 310	83 259	84 602

Figure 41: The classifier map shows, how the labelled classes are located on the map after retraining. The first number in the class is the class index, the second one is the number of training samples and the third one gives the index of the respective class on the previous classifier map. It can be seen that a new set of data was used to train the classes in the top left corner of the map.

The classes on the new map have been labelled by searching for the best matching class for each of the memorized weight vector. The class has then been labelled accordingly. Because of the weight vectors represented the average symptom values of the training samples, very high membership was detected. Some interesting observations can be made:

1. There were no hits in data set “GB02” in the classes 29 to 32 that represented the most severe fault mode in data set “GB01”. See distances of data samples in these classes in Figure 42.
2. There were no hits in data set “GB01” to classes 21 to 28 that represented the most severe fault mode in data set “GB02”. See distances of data samples in these classes in Figure 43.
3. Both data sets have hits in classes 53 to 56 before entering the classes mentioned in previous points.

The first two points above verify that the fault modes are different at least at the later stage of the fault progression. In fact, the bearing fault in “GB01” was in an outer race of the bearing, while in “GB02” it was in an inner race. Regardless of the differences between weight centres in classes 21 and 29, the two fault modes are still in the neighboring classes. As a conclusion, it is difficult to predict the correct type of an early bearing fault, when we are still in classes 53 to 56, where the state of both gearboxes remained for more than six weeks.

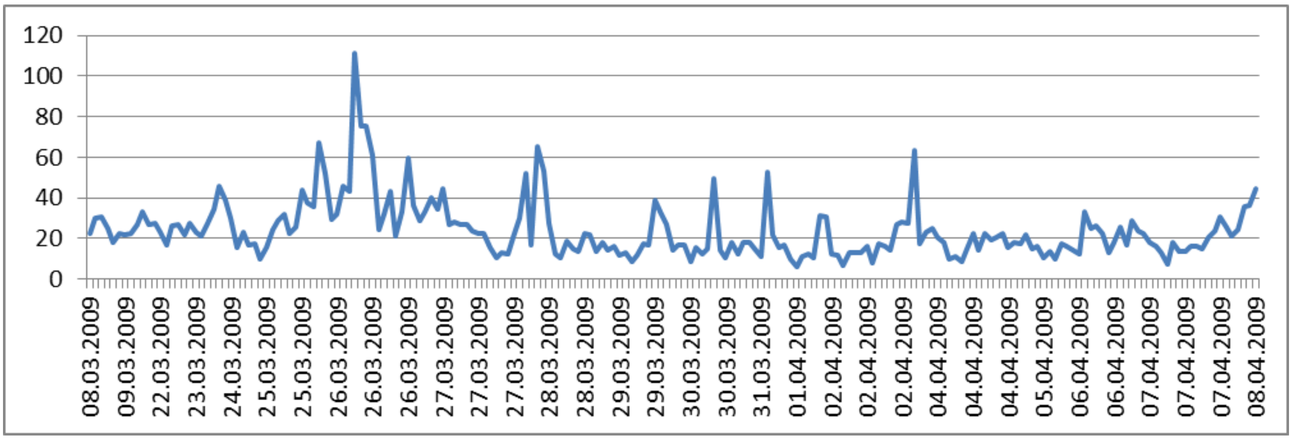

Figure 42: Euclidean distances of data samples from respective weight centres in classes 29 to 32 representing the most severe fault mode of data set “GB01”.

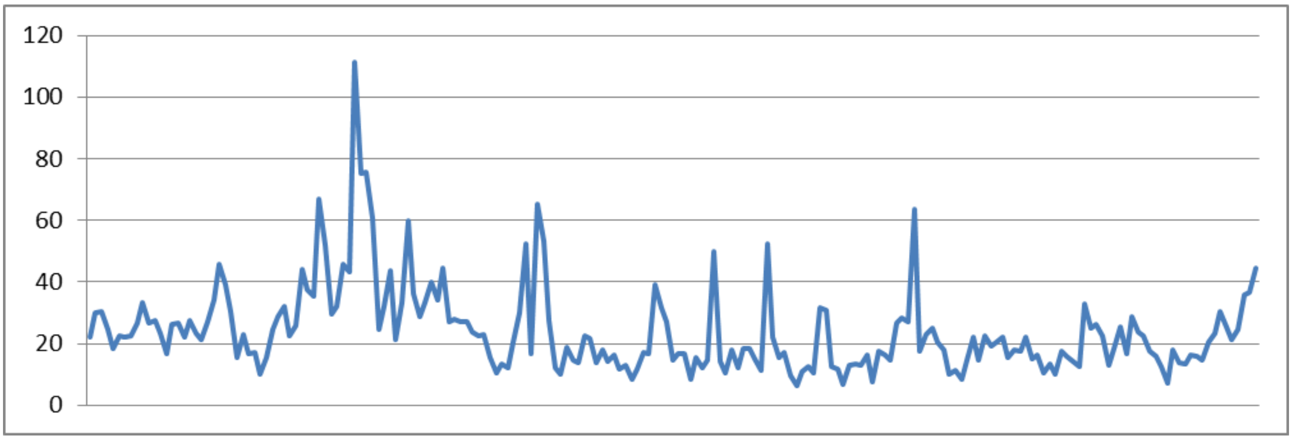

Figure 43: Euclidean distances of data samples from respective weight centres in classes 21 to 28 representing the most severe fault mode of data set “GB02”

This experiment proves that we can use a classifier to detect anomalies that might be caused by developing faults. We can also differentiate closely resembling fault modes even, when they appear in different machines in the same environment. The next challenge is to generalize the anomaly and fault detection even more.

6.7. Generalization

In conventional CM systems, the machines are monitored by trending the descriptor values and comparing them against the baselines based on data from the machine itself. It would be beneficial, if a learning system could utilize the data collected from a large amount of similar or substantially similar machines that have experienced various fault modes. The faults are generally rare and all information and knowledge thereof is valuable. [Lumme, 2003]

For the next experiment data was selected from a wind turbine gearbox “GB03” in Germany. A total of 871 data samples were available after data pre-processing. The number of hits in various classes is illustrated in Figure 44. It appears that all data falls in a map section, which is considered to represent the normal modes.

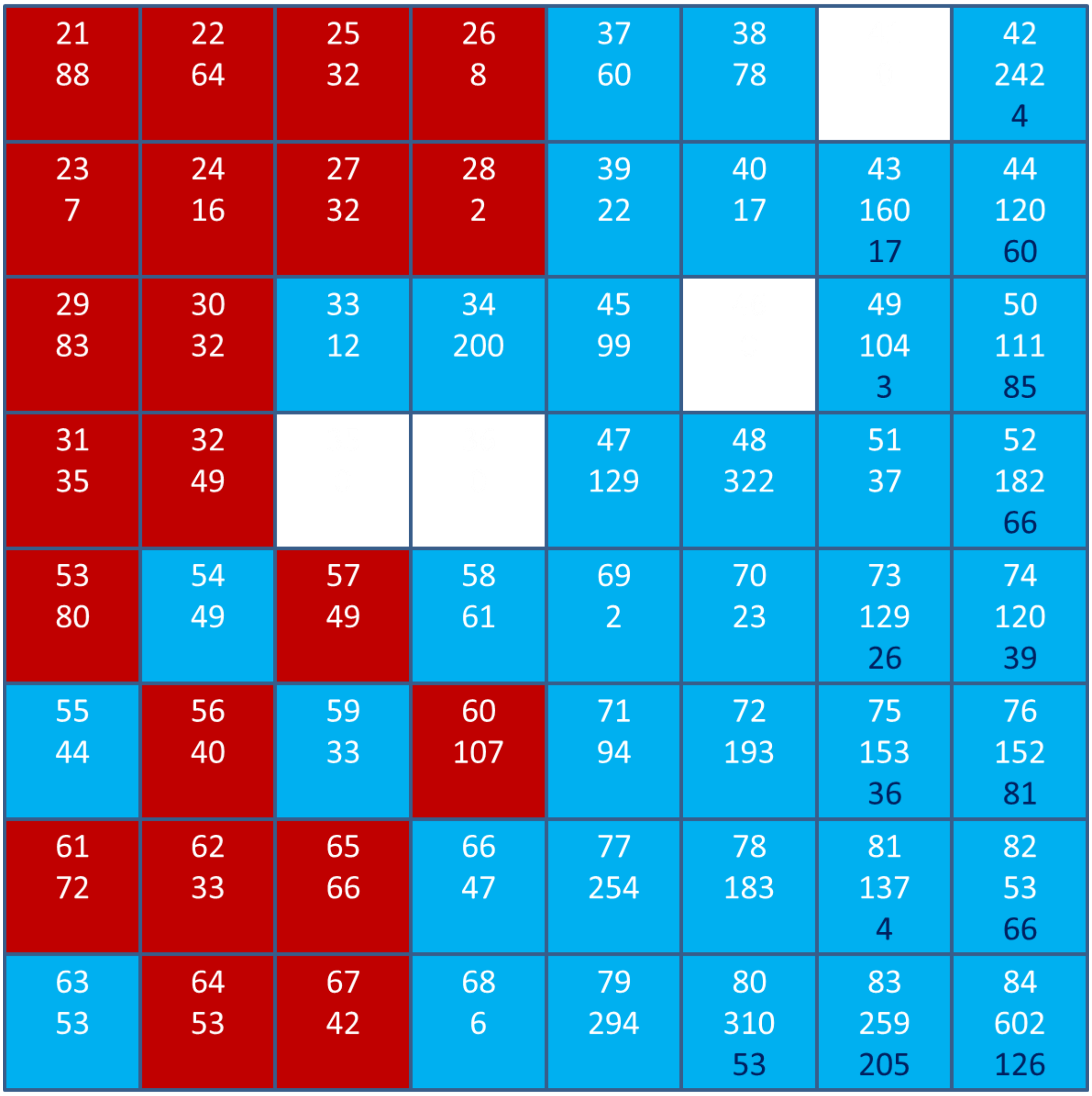

Figure 44: The number of hits (third number in a class) of a data set “GB03” in a classifier map trained with data sets “GB01” and “GB02”. The classes with red background mark fault modes. All hits are in the left side of the classifier map representing normal modes.

The logarithmic presentation of anomaly index versus time is shown in Figure 45. The gearbox was found to be in a good condition, which can be seen in a very high membership for most of the data samples. There are, however, a few sporadic anomalies, when data samples hit classes 44, 50 and 73, where the current maximum distance is very low. This was a result of many NaN values in the descriptor values of the original training data. This means that even a small deviation from the mean descriptor values causes an anomaly. It would be safe to append the anomaly data unlabelled (as a normal mode) into the training data for retraining. This experiment proved that the data collected at two distant machines can be used as a baseline for prediction of data taken from a third similar machine.

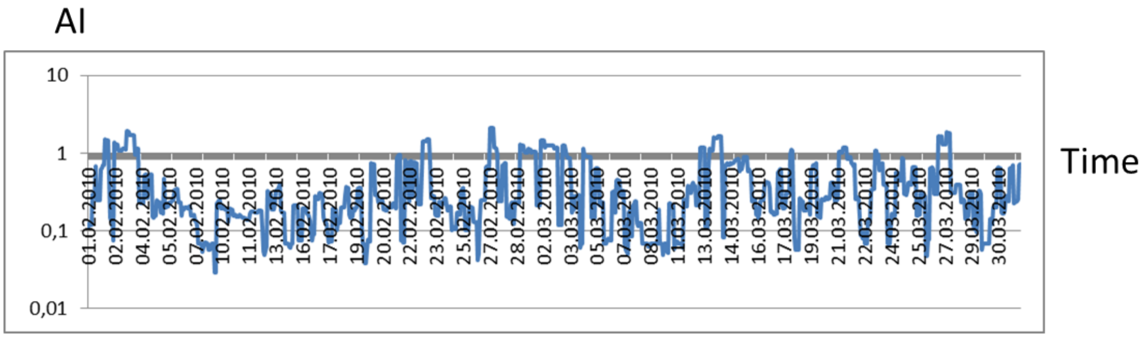

Figure 45: Anomaly index trend shows high membership of most data samples in the current classifier proving that the data collected from one gearbox is almost equal to data collected from another gearbox.

We need more data to prove the common principle of generalization. For this purpose data was selected from 19 wind turbine gearboxes (including the ones already discussed in this chapter) in four different wind turbine parks. After data pre-processing 20781 data samples were available for training. The whole data population was used to calculate the means and standard deviations for normalization. Figure 46 illustrates the classifier map after initial training. The classes linked to the two known fault modes have been marked accordingly. Note that due to the powerful search engine of the TS-SOM, the training process took only less than 30 seconds.

21 GB02	22 GB02	25 GB01 GB02	26 GB01	37 GB01 GB02	38 GB01 GB02		42 GB01
23 GB02	24 GB01 GB02	27 GB02	28 GB01	39 GB01 GB02	40 GB01 GB02	43 GB01 GB02	44 GB01 GB02
29 GB01	30	33 GB02	34	45 GB02	46 GB01 GB02	49 GB01 GB02	50 GB01 GB02
31 GB01	32 GB02	35	36	47 GB02	48 GB01 GB02	51 GB01 GB02	52 GB01 GB02
53 GB01 GB02	54 GB01 GB02	57	58	69 GB02	70 GB02	73 GB02	74 GB02
55 GB01 GB02	56 GB02	59 GB01 GB02	60 GB01 GB02	71 GB02	72 GB01 GB02	75 GB01 GB02	76 GB01 GB02
61 GB01	62 GB01 GB02	65 GB01 GB02	66 GB01 GB02	77 GB01 GB02	78 GB01 GB02	81 GB01 GB02	82 GB01 GB02
63 GB01	64 GB01	67 GB01	68 GB01 GB02	79 GB01 GB02	80 GB01	83 GB01 GB02	84 GB01 GB02

Figure 46: Classifier map after initial training shows the distribution of training data samples in various classes. Classes marked in blue have no hits, yellow hits from data set “GB01” only, red hits from data set “GB02” only and orange from both data sets.

It appears that the data from these gearboxes spreads out evenly on the classifier map. Almost all of the classes are used to represent the various states reaching from low symptom values in normal modes to high values in fault modes. This can be understood in a way that generalization is effective. On the other hand, the map might give an impression that the classifier is no longer sensitive to differentiate between fault modes or severities.

6.8. Improving the classifier

As the data volume increases and especially, when new fault modes are encountered, it may seem that the classifier becomes overloaded. In other words, it looks like there is no space left for new totally different fault modes. We have already seen that the classifier adapts itself to the new data, but eventually it will lose some of its sensitivity, as data samples with higher differences get classified into the same class. When the hypersphere of anomalies increases, the main concern is in the anomaly detection.

There are several methods to improve the situation. The most obvious solution would be the increase of the number of classes. Figure 47 presents the same data as in Figure 46, but in a five layer classifier with a maximum of 256 active classes.

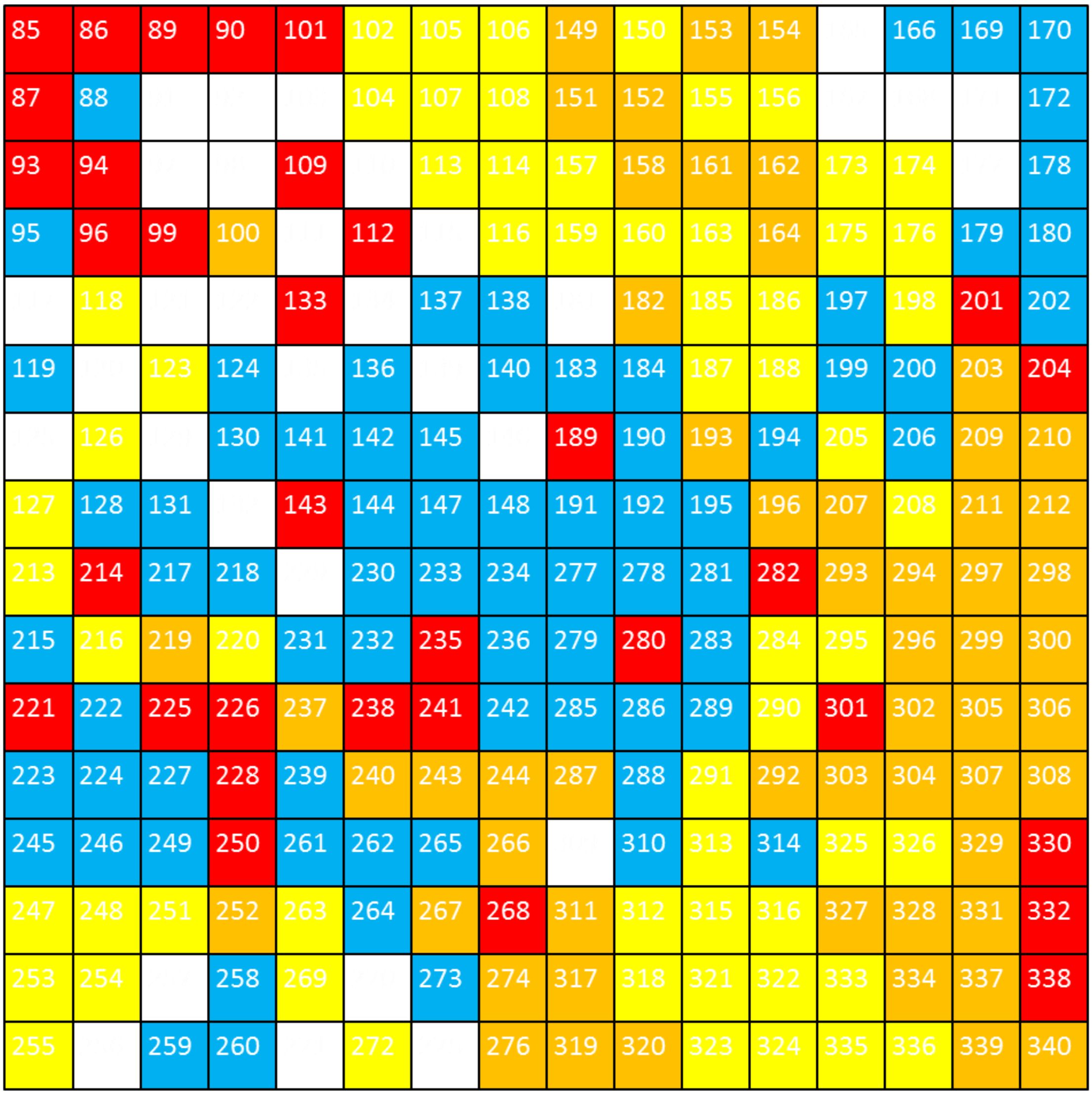

Figure 47: Five layer classifier map offers improved sensitivity to differentiate between various types of data. The colour coding is the same as in the previous figure.

With a greater number of classes and better sensitivity, the diversity between the two failure modes can be observed at the top left corner. Also, it is clear that there are other data samples representing fault modes in the training data. The disadvantage of using more classes is the increased number of search processes, which results in slower performance. Actually, we are only interested in the improved sensitivity in separating between fault modes, which in this case are represented in the classes close to the top left corner. Figure 48 shows a classifier that was created based on the 2019 samples that hit classes 21 to 36 in Figure 47. The classifier in four and five layers is shown. Data was not re-normalized using the restricted data population before training. Because of the tree-structured SOM, we do not necessary need to search down to the forth layer to find the data samples linked to fault modes, but we can select only the 3561 data samples hitting top left class in the second layer for re-training.

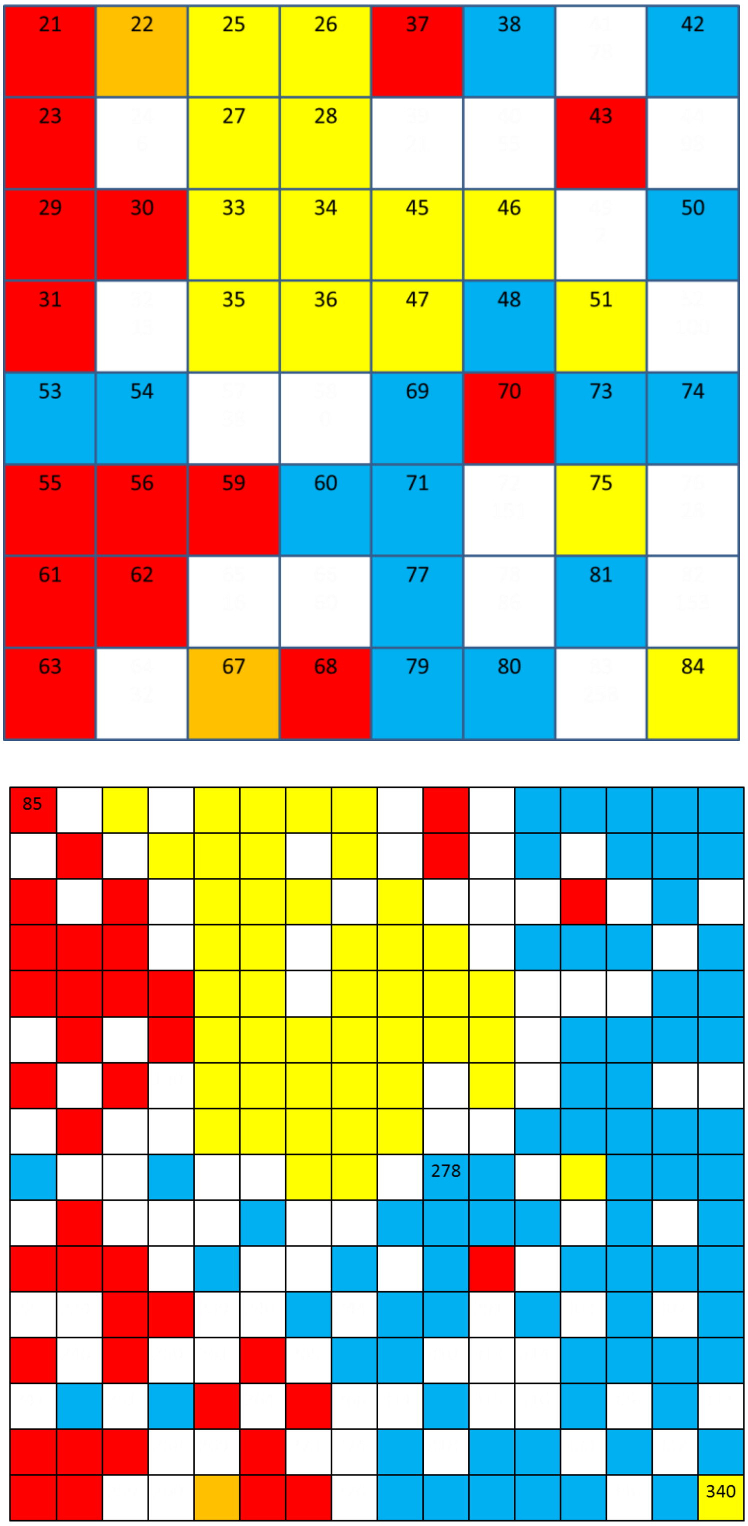

Figure 48: Four (upper) and five (lower) layer classifier maps trained with 3561 data samples that hit the classes in the top left quarter of the classifier in Figure 47. Colour coding is maintained the same as in the previous figures.

Some observations can be made on these maps. As the sensitivity has been further increased, more unlearned classes appear between two fault modes. The class regions or clusters marked with labels and colours can be distinguished clearly. According to this classification there seems to be no common progression path between the two modes and the fault in “GB02” is obviously more severe at least considering the symptom magnitudes.

Classes 21 (in upper map) in Figure 48 and 85 (in lower map) have been trained with data samples collected from a wind turbine gearbox “GB04” in Australia. By investigating the symptoms of the data samples, it can be confirmed that the gearbox “GB04” is in fact experiencing the same fault mode as gearbox “GB02”. This experiment proves that a particular fault mode can be detected, if first trained with the data from another wind turbine on the other side of the globe.

The use of a higher layer classifier apparently provides a significant improvement in the classification sensitivity, but it requires more processing power. Because of the tree-structured SOM this added processing is not tremendous, but in some cases it is desirable to delimit the number of classes, especially if classification is planned to be performed in microprocessors.

Looking at the weight vectors of the classes in the bottom right quarter of the map in Figure 48, we can see close similarity. The distance between the weight centres of the furthest classes 277 and 340 is 33.5. We might consider replacing all these classes with a single class on the second layer with a maximum distance of 2870. The distance of weight centre in the new class to those in classes 277 and 340 are 20.1 and 5.4 respectively. All these distances are reasonably short compared with those in the other quarter. The map would then look like the one in Figure 49.

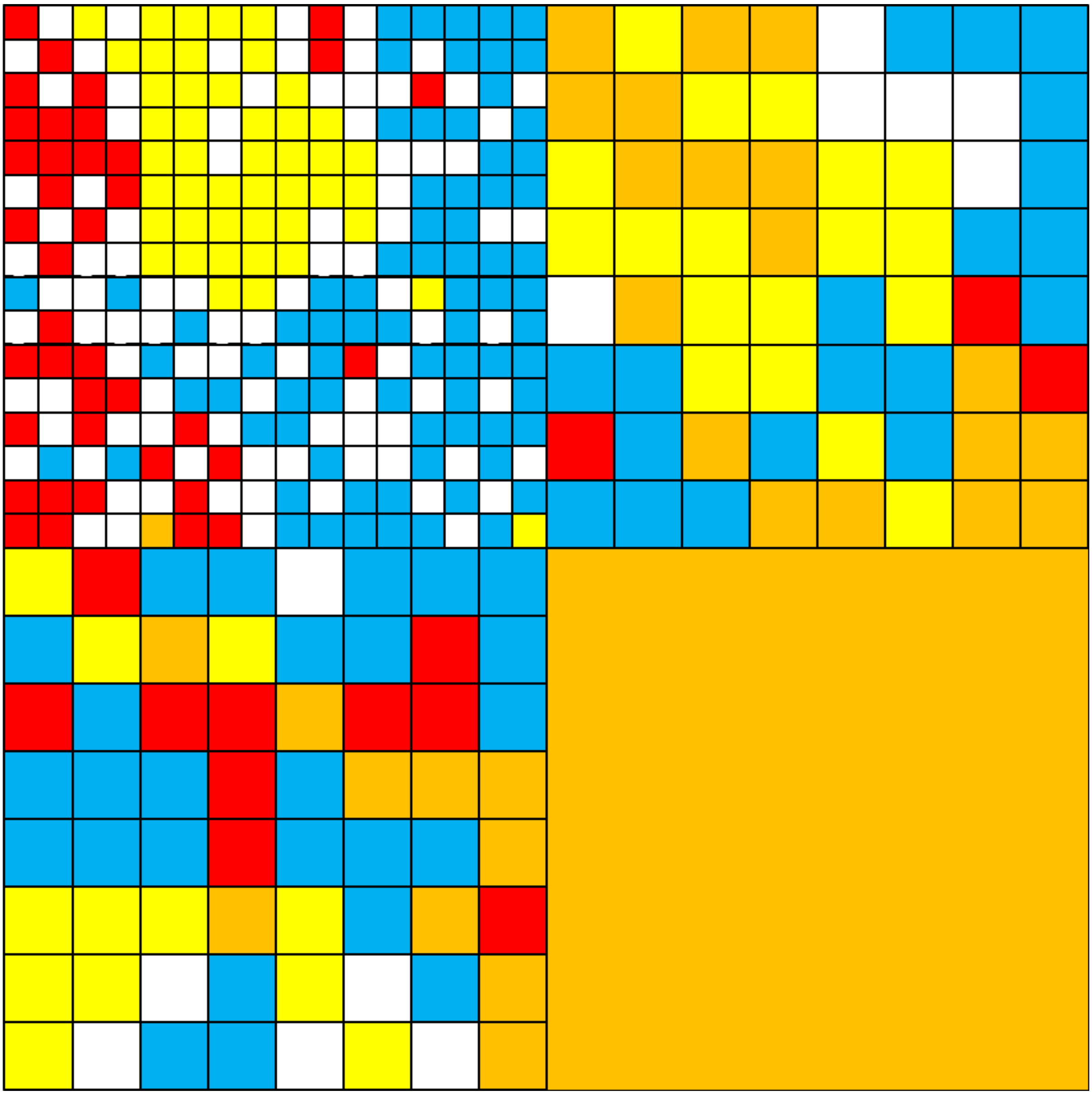

Figure 49: Combined classifier map uses more classes in the top left corner to represent the data with high symptom values and only one class in the bottom right corner to represent normal modes.

The classifier now has about 270 active classes, where of 160 are high sensitive for detection of fault modes. It should be noted that the class 4 could be smaller and larger depending on the data at the bordering classes in the top right and bottom left quarters.

With this clustering method we are taking a risk in missing the detection of some anomalies. Let us test the classifier with some data samples that hit the previous classes between 277 and 340. Nine data samples with the longest distances from their class weight centers were selected for prediction and the results are summarized in Table 2.

Table 2 Comparison of class distances

Sample	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Old class	332	279	279	279	279	308	279	279	279
Old distance	106	125	125	125	125	132	825	825	825
New distance	132	149	149	149	149	136	884	884	884

The test shows that the distances for the furthest data sample from the new weight centre are longer. This means that new radius of the hyper sphere used for anomaly detection will be slightly longer than the old one and the classifier will be less sensitive. In other words, some anomalies related to new fault modes at the class borders might remain unnoticed. The possibility is, however, negligible.

6.9. Evaluation of the confidence level of prediction

A simple method to test the confidence level of prediction is to train the classifier and predict the outcome with the same data set. Obviously there should be an equal amount of training samples and predicted samples in each class. This is not always the case. Training of a classifier is an iterative process that is interrupted, when pre-determined classification accuracy is reached. It may happen that the class centres are not at the ideal locations in the hyperspace after training. Especially, when the TS-SOM algorithm is used, the best matching class is searched among the descendants of the parent class during training. The prediction algorithm might use another, more accurate method, where the best matching class is searched among all the classes on the lowest layer. As a result, the best matching class might be found in the neighbouring class. In a practical application, this is not a real problem, because the neighbouring classes typically represent data with only minor variation. In case of a class representing a fault mode, the neighbouring class probably represents the same fault mode in a slightly different severity. [Lumme, 2012b]

In the first test, the classifier has been created using the four layer TS-SOM algorithm with 64 classes on the lowest layer. The prediction has been performed by searching the best matching class among the 64 classes on the lowest layer. Table 3 presents the results. The first column gives the reference to the data set and the second column the number of data samples used for training and prediction. The next column gives the number of data samples used for prediction that did not hit the same classes during training and prediction. The percentage in the fourth column has been calculated by dividing the number of differing samples by the number of training samples.

Table 3 Prediction error

Data set	Amount of samples	Amount of differing samples	Error percentage
GB01 & GB02	2690	0	0 %
GB03 & GB04	3150	0	0 %
GB05 & GB06	2627	0	0 %
GB07	2676	2	0.075 %
GB08, GB09 & GB10	2711	0	0 %
In total	13854	2	0.014 %

The results given in Table 3 show that only for two data samples out of 13854 the predicted class was other than expected. In this case, the two samples hit a neighbouring class, which is not a major concern. The conclusion is that there is a high probability that the prediction by the classifier is correct.

When the same data set is used for training and prediction, the number of anomalies, i.e. novelties, should be zero. All data samples should fall into the limits of borders of the best matching class.

Table 4 Error in anomaly detection

Data set	Amount of samples	Amount of novelties	Percentage of novelties
GB71 & GB72	2690	25	0.93 %
GB73 & GB74	3150	19	0.60 %
GB75 & GB76	2627	19	0.72 %
GB77	2676	22	0.82 %

GB78, GB79 & GB80	2711	25	0.92 %
In total	13854	110	0.79 %

Table 4 summarizes that 0.79% of all data samples caused undesired anomaly detection. In this case a sigmoid membership was used to define the class borders. No tolerances for exceeding the class borders were used. The error in anomaly detection was caused by extremely minor crossings of class borders. In a practical application, the class border is often fuzzified to allow minor crossings. We may conclude that there is a high probability that there are no anomalies in the data samples.

The confidence of the classifier to predict various normal modes of operation is highly dependent on the amount and variation of data samples used for training. In a long run, the system should be able to effectively detect all normal modes of operation, but also react to any anomalies, which might possibly result from a fault condition. For the next experiment, the initial training data was collected from five wind turbine gearboxes in different wind parks. The prediction was done with five datasets from the same wind parks, but different gearboxes than those used for training. Both the training and prediction samples did not include any data from known fault conditions. The number of novelties within a specific period of time should not be excessive compared with the total number of interpretations. However, as the condition typically stays unchanged for several successive measurements, the number of detected anomalies is not important, but the duration. The limit for anomaly detection was selected as 30 % meaning that the class border could be exceeded 1.15 times the maximum distance without being an anomaly.

Table 5 Anomaly detection

Data set	Samples	Anomalies	Percentage
GB10	1371	78	5.7 %
GB11	1407	377	26.8 %
GB12	1386	30	2.2 %
GB13	1831	72	3.9 %
GB14	1243	697	56.1 %
Total	7238	1254	17.3 %

Table 5 shows that there are quite many anomalies. Most of these are single events and can be considered as outliers. In a practical application, the amount of detected anomalies should be less than five per cent of the total amount of normal data samples for a detection to be effective. In this experiment, this limit is exceeded significantly. The conclusion is that the anomaly detection should be improved to ignore the outliers.

The classifier should be able to detect any fault condition that results in a change of descriptor values, i.e. causes symptoms. Clearly, the classifier cannot predict a fault mode after the initial training with normal unlabelled data samples, but should indicate an anomaly. This property can be demonstrated and measured by first training the classifier with representative data samples from normal conditions and then testing with data samples representing a known fault condition. The system should detect a fault before or latest at the same time as a human expert. Table 6 summarizes the results.

Table 6 Fault detection system versus analyst

Dataset	Anomaly detected by the classifier	Fault reported by an	Difference in days
GB20	22.11.2009	18.12.2009	26
GB21	5.5.2010	7.5.2010	2
GB22	27.1.2010	8.2.2010	12
GB23	9.3.2010	21.4.2010	43
GB24	25.1.2010	12.2.2010	18
Average			20

The training data used in the experiment was the same as in the previous tests. The experiment reveals that in all cases the classifier could detect the anomaly earlier than reported by an analyst. In the best case, the warning was given 43 days earlier. The conclusion is that a fault can be detected as an anomaly certainly.

The ability of the classifier to predict data samples representing normal condition correctly was tested with data sets from 24 wind turbine gearboxes at various locations. The training data represented mostly normal modes, but included also data from various fault modes. It is desirable that none of the data collected from gearboxes running in a normal mode would be predicted to belong to a class representing any of the fault modes. The experiment showed that all data collected in normal mode was predicted to belong to the "normal class". None of "normal data" was predicted to any "faulty class". As a conclusion it is certain that normal modes can be predicted accurately.

The classifier should be able to predict a known fault mode. This ability can be demonstrated by introducing to the classifier a set of data samples from a different, but similar fault conditions to the one already used for training the classifier. It should be able to predict the data samples into the respective classes with or without anomaly features.

The last experiment consists of data collected from 22 wind turbine gearboxes at various locations. Each of the data sets includes data from fault modes. They are all bearing faults that usually lead to immediate actions. The data sets in Table 7 marked as bold (1, 2, 3, 9, 10, 11 and 21) were used for training the classifier. There were two basic fault modes in the data set: IR marks a bearing fault on an inner race and OR a bearing fault on an outer race. In addition the data set No 7 includes data from a bearing fault in a planet wheel. The data set No 9 consists of data collected from an unknown fault mode.

Table 7 Confidence level of fault prediction

No	Data set	Fault mode	Predicted by classifier	Reported by analyst	Difference in days
1	**GB40**	IR	30.12.2010	8.1.2010	9
2	**GB41**	OR	20.9.2010	6.10.2010	16
3	**GB42**	IR	20.11.2009	23.11.2009	3
4	GB43	IR	8.5.2010	10.5.2010	2
5	GB44	IR	25.11.2010	26.11.2010	1
6	GB45	OR	25.10.2009	18.12.2009	54
7	GB46	Planet wheel bearing	a)	14.12.2010	
8	GB47	IR	21.8.2010	23.8.2010	2
9	**GB48**	OR	b)	27.9.2010	
10	**GB49**	Undetected	28.11.2009	c)	
11	**GB50**	IR	2.5.2010	7.5.2010	5
12	GB51	OR	25.6.2010	29.6.2010	4
13	GB52	IR	d)	3.6.2010	
14	GB53	IR	4.9.2009	18.11.2009	75
15	GB54	OR	10.5.2010	17.5.2010	7
16	GB55	IR	18.12.2009	5.1.2010	18
17	GB56	IR	29.1.2009	14.1.2010	14
18	GB57	IR	b)	27.7.2010	
19	GB58	IR	18.9.2010	21.9.2010	3
20	GB59	IR	1.2.2010	8.2.2010	7
21	**GB60**	IR	b)	25.2.2011	
22	GB61	IR	16.7.2010	20.9.2010	66

The confidence level of fault prediction has been analysed by comparing the date, when the fault was first predicted against the date, when an analyst reported the fault. It should be understood that the analyst might have detected the fault a day or two earlier than reported, but on the other hand the classifier may be deemed to have already reported a fault, when it was first predicted. In some cases the classifier did not predict the fault mode, but detected an anomaly instead.

This comparison is visualized for data set number 22 in Figure 50. The time line shows anomalies in blue and fault modes in yellow, orange or red depending on the severity. The white colour marks unavailable data due to for instance a measurement error. It appears that the fault mode has first been predicted 16.7.2010 (in yellow). After that a series of anomalies can be observed. This application visualizes the anomalies (in blue) before the fault modes. It would be preferable to show the fault mode with a higher priority. Also it should be noted that the gearbox might move away from the "fault mode" class, when the speed or load changes. This is likely to happen several times in wind turbine gearboxes within the two months' time span shown in the Figure 50.

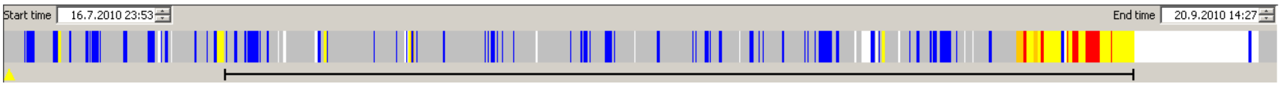

Figure 50: The time line shows anomalies in blue, moderate faults in yellow and severe faults in red. The white sections indicate that data is not available, because of a measurement error or machine being out of operation.

The following notes apply to Table 7:

a) The classifier did not predict the correct fault mode, because such data was not used to train the classifier. The fault mode was detected as a significant amount of anomalies.
b) The classifier did not either detect an anomaly or predict a fault mode.
c) The fault mode was not reported by the analyst.
d) The fault mode was not clearly predicted. Anomalies could be detected, but their number was not significant.

In summary, four occurences of fault modes could not be either detected as anomalies or predicted. At its best the classifier predicted a fault mode 75 days before the analyst. The classifier therefore predicts in a high probablity a fault mode that has been encountered before.

7. PROCESS STATES IN CONDITION PREDICTION

7.1. Process state

In vibration monitoring of rotating machines, the conventional diagnosis and prognosis of a machine condition relies on a trending of sequential measurement values. The process state may have a great impact on the vibration response of the item. In condition monitoring, it is usually assumed that successive measurements have been recorded under comparable states. This is often achieved by timing the measurements so that the process is in a substantially same state as during the previous measurements. This is not always possible or practical. Especially, the automated monitoring systems collect data at fixed intervals regardless of the process state. The conditions, under which the measurements are recorded, are not always known to the analyst. If this is not taken into account, an erroneous diagnosis or prognosis may result. [Lumme, 2005]

It is therefore very important for the vibration analyst to know, in which state the process was during the collection of vibration data. The individual process input and output values as such are generally not important, but the process state in general. Instead of the actual process output values there might be other types of condition related data available that could be used to confirm the diagnosis performed on the basis of the vibration data. Another data type might in some cases contain even more valuable information than the vibration data. Again, the individual data values might not be of interest, but the different states of conditions. The correlation between the process data and condition monitoring data has already been studied in the Laboratory of Machine Dynamics at Tampere University of Technology. The study revealed that cause-effect relations can be found between the different data [Setälä, 2004].

The determination of a process output state is based on an evaluation of characteristic descriptors or symptoms in the process data. By using known rules or learning methods, these symptoms can be interpreted as different operational states. The state might for instance be evaluated as normal, deviating or unfavourable. If the state is determined to be abnormal, it is important to find out the characteristics, through which the sample deviates from the other ones. Similarly, if the state is unfavourable, the process should be examined to determine the cause and duration of the behaviour before harmful consequences to the process or manufactured product occur.

An industrial process is controlled by one or more input parameters. The combination of input values related to a specific process or sub-process constitutes an input state. There can be numerous different possible input states for a single process. Consequently, the process is monitored by one or more process output values. The combination of output values can be called the output state. The purpose of process operation is to select a proper input state to produce a desired output state.

The process output state can here be understood in a broad meaning. In addition to monitoring of the process output parameters, secondary results of the process can also be recorded. For instance, the process productivity, the vibration state of rotating machines and the stresses in a structure can be secondary output states, whose desired objectives can be defined in advance.

Figure 51 illustrates a process as a "black box" that can controlled by a combination of input variables, the values of which concurrently form an input state. In a similar manner, the output values form an output state. Obviously there are numerous possible input and output states for a single process. Some of them are favourable and some should be avoided. [Lumme, FI118746]

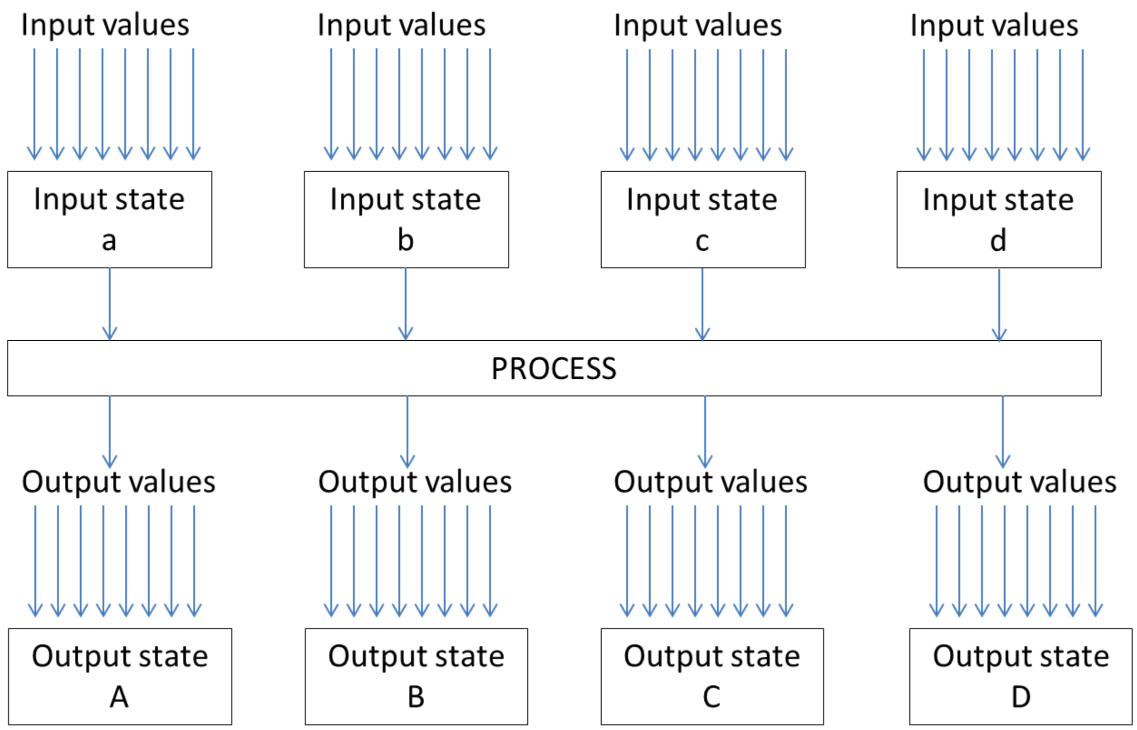

Figure 51: Process can be seen as a black box, where a desired output state is achieved by setting the appropriate input state. The input and output states are combinations of respective input and output values.

7.2. Determination of process classes

As there are numerous different input and output states for a single process, it would be difficult to find the relationship between the respective states. In cases, where the output state is not a direct indicator of the process outcome itself, but for instance the vibration behavior of the process equipment, it would not be rational to investigate the relationship between the input and output states. Small variations in input values change the input state, but this might have no effect on the output state. Similarly, small changes in output values might not be a result of a changed input state. In an attempt to find the relationship between the states, small variations should therefore be allowed.

The previous chapters have presented methods to utilize classes as representatives of several closely related data samples. In a same way, input and output classes can be formed to represent the combinations of input and output values. The determination of the relationship between input and output classes using learning methods can be accomplished by first training input and output classifiers and then by detecting the changes in the input and output states.

The same classification principle that was described in Chapter 4 can be used handle input and output data. Data sets consisting of input values are used as training data for a input SOM classifier, which creates input classes I_1 to I_n. Clearly, there are far less input classes than input states, as they represent the groups of similar input states. Similarly, the process output can be monitored by output values, which form various output states. These can be used as training data to another (output) SOM classifier, which creates output classes O_1 to O_n. See Figure 52.

The relationship between the input and output class is not always clear, but by using previous experience some input classes can be marked unfavourable or harmful, because they are known to lead to disadvantageous process classes. On the other hand, some process classes might be favourable. The process operator should thus select a suitable input class I_i to reach an ideal output class O_i.

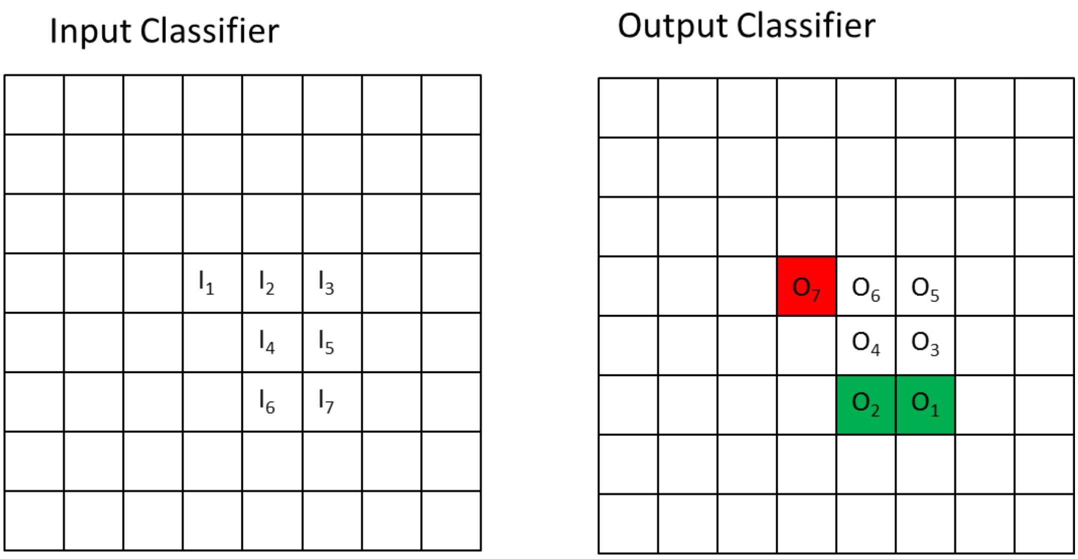

Figure 52: Two separate classifiers have been created using input and output training data respectively. Output classes O_1 and O_2 have been marked to represent favourable states and class O_7 an avoidable state.

Anomaly states that have not been used for training either of the classifiers can be detected, when membership of a data sample in the best matching class is low. The data from the anomaly can be used for retraining. The continuous learning mechanism can be applied also in the classification and prediction of various input and output states that were not encountered during the initial training.

7.3. Determination cause-effect relationship

A cause-effect relationship should exist between the control state and the output state. It is not always clear, which change in an input state causes a required change in an output state. An input state might change without a change in the output state. In addition, there is typically a delay between the states, the duration of which is not known.

Figure 53 shows the principle of finding the cause-effect relationships of a process. At the initial moment the input state is in class I_1. At the same moment output state is in class O_5. While both the input and output states are steady or vary less than would cause a change to another class, there is probably a relationship between classes I_1 and O_5. One might even be able to conclude that the input class I_1 causes the output class O_5. Due to the change of the input state by setting of the input parameters, the output class shifts from I_1 to I_7 and after a delay the output class shifts from O_5 to O_2. After becoming stable, there is a relationship between classes I_7 and O_2. Consequently, we have found two potential pairs of relationships.

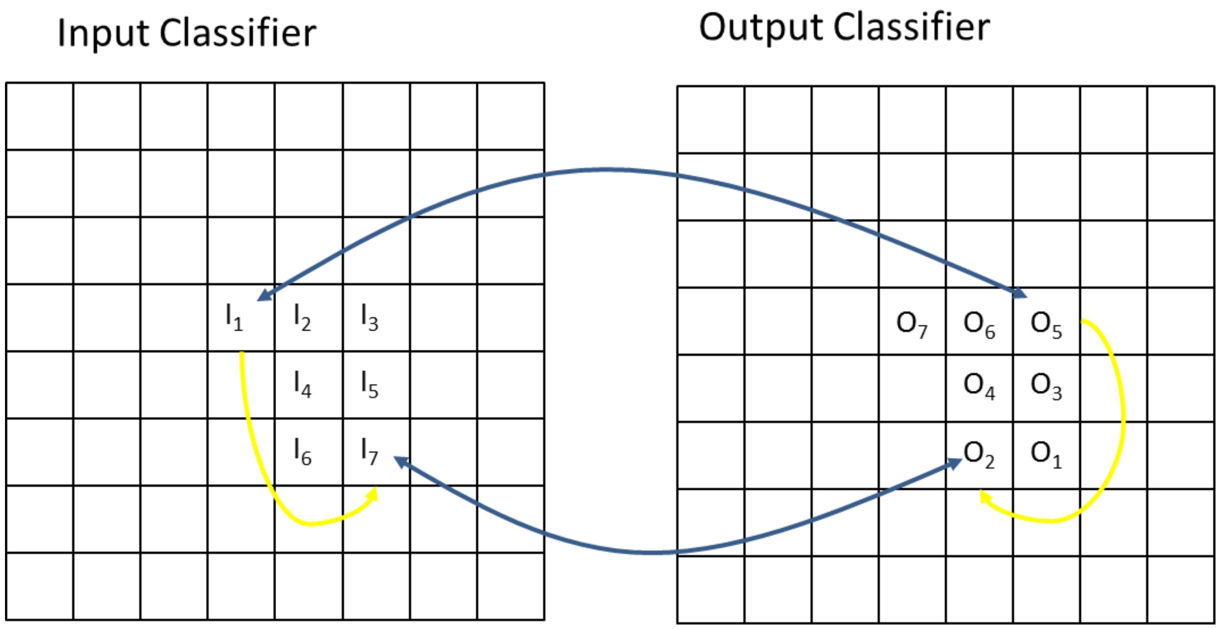

Figure 53: Cause-effect relationship between control and process states can be found by linking an input class with an output class in steady process conditions.

By further adjustment of the process parameters, more classes can be trained and predicted. As a result, more potential relationship pairs will be found. It may turn out that some of the pairs are repeatable, while others are not. If they are repeatable, the confidence level of the pair being real is high. Otherwise, the cause-effect relationship is unsure and changes might be caused by other factors. In case a repeatable relationship is found, we should evaluate, if the process class is favourable or harmful. The respective control class should be labelled accordingly. Let's assume as an example that class O_5 is favourable and class O_2 is uneconomical to the process. The respective control classes could be labelled with the same interpretations. If a process operator is to change the control parameters, he or she might get a warning that we are about the leave a favourable process state. If the operator is further changing the control parameters so that control class I_7 is reached, he or she might get a warning that continuation of process operation with these control parameters will lead to an uneconomical process state after a certain delay.

7.4. Prediction based on the process states

How does this all connect to the prediction of machine condition? We have learned above that it is possible to predict an output class by knowing the input class. It is also known that the process state has an effect on the vibration state, which can in this case be understood as an output state. When this relationship is discovered, the condition monitoring system should expect to "sense" a certain vibration state. Let's use a simple example to demonstrate this prediction. Figure 54 illustrates a classifier trained with vibration data collected from a wind turbine gearbox known to be in a good condition. It is common knowledge that the vibration response of such a machine is strongly dependent on the rotation speed. As no other process parameters were available to concurrently determine the process state, the rotation speed alone is used here for the prediction.

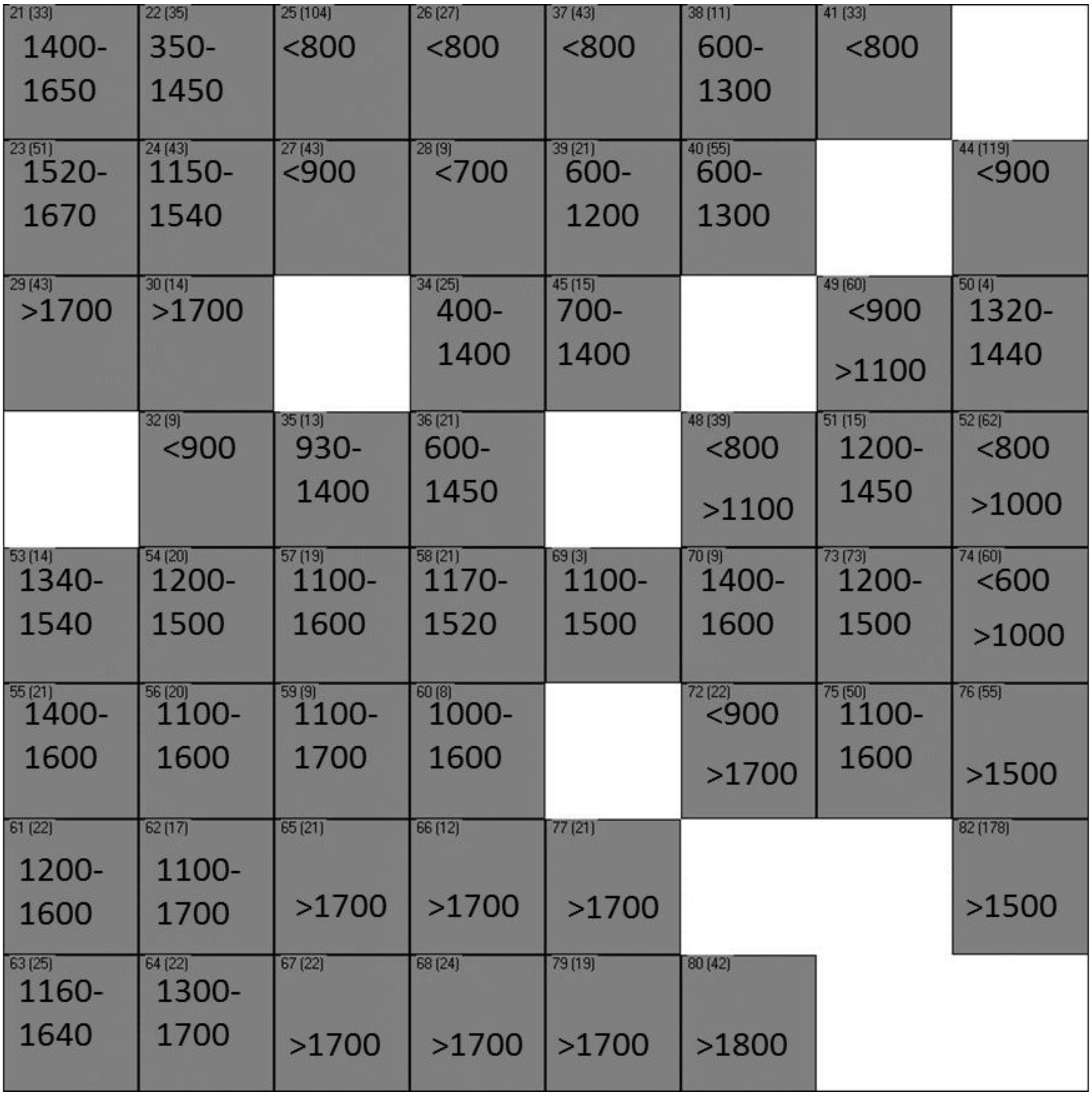

Figure 54: Training data samples have been collected at various rotation speeds. Classes in this classifier map have been marked with the speed ranges of the data samples used to train a particular class. In most classes there is a large speed range among the training samples. For instance, when the rotation speed is over 1800 rpm, the data is expected to belong to class number 80.

Figure 54 shows that classes on the map are not strongly dependent on the rotation speed. In fact, it is difficult to draw conclusions based on the rotation speed. However, the information gained in this example can be used in some extent to predict the data to be expected. For instance, if the rotation speed is higher than 1800 rpm, we should expect to collect data belonging to class number 80. If the classifier predicts any other class, this should be considered as an anomaly. If the rotation speed is less than 1800 rpm, but over 1700 rpm, there are much more possible predictions including classes 29, 30, 65, 66, 67, 68 and 72.

This experiment shows that rotation speed as a single descriptor is not adequate to define the vibration state. Probably, load would be a more sensitive descriptor of machine behaviour and should be used jointly with rotation speed to form the input state. It should be noted, however, that these descriptors might have different temporal meanings.

8. FUTURE DEVELOPMENTS

The previous experiments bring forth several ideas and improvements. The main focus should be in the development on the user interface. However, there are also aspects related in the training process.

For the purpose of demonstrating the system's ability to handle large number of symptoms, the significance of each symptom was not investigated in this study. With several symptoms being used it is likely that one or more symptoms are either correlated or not representatives of any fault modes. There are several methods, such as principal component analysis (PCA), to evaluate the significance of each symptom.

The usage of unsupervised Self-Organizing Map as a classifier requires that all training data need to be saved. The training databank will continue to increase in size as new anomalies and fault modes are appended. Because faults are typically rare, most of the training data is likely to represent various normal modes. The speed to process the data used in the experiments was reasonable, but in some point it may become intolerable. There are possibilities to manage the size of the training databank. For instance, more than 11000 data samples have been used to train normal modes in the experiments. The same accuracy and confidence of prediction would most probably be achieved by using only 100 data samples or less. A supervised SOM [Kohonen, 2001] or some other supervised learning method may also prove useful in minimizing the amount of training data to be maintained.

This study was based on an assumption that the vibration state of machines is independent of their age or operating hours. In reality, the failure rate and probability will generally increase along with time as the machine wears on. If the excitation forces, such as those caused by imbalance or misalignment are low, the wear is slow. Wear may progress extremely fast, if proactive actions are not taken to minimize the excitation forces. Conventional condition monitoring relies on knowledge that the prevailing machine condition can be confidently determined by analysing condition related data, but in reality the failure pattern and progress might vary, if the fault is caused by excessive excitation forces, process failures or manufacturing faults of a machine component. Further studies should be made to differentiate between the different between failure patterns in order to accurately predict the remaining life of a machine. This may involve the use of a forgetting factor to emphasize the most recent and fault mode related training data.

The experiments proved that the search algorithm for localization of the best matching class is very fast, when using a tree-structured SOM even in the iterative process of the training. However, the TS-SOM search process requires more classes to be saved. For example, a five layer classifier creates 340 classes, whereof only 256 classes are used for prediction. In applications, where memory management is a concern, this might set restrictions. On the other hand, the search process does not need to be fast in the prediction of samples. It is more important in the training process, which involves several iterative search processes. In fact, the prediction process could easily be implemented in a microchip with a price less than 20 euros. For instance, Arduino offers an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software in order to build a prediction system [Arduino, 2012]. Another choice is offered by Digilent [Digilent, 2012].

Processing the prediction in a microchip might require that the symptoms are given in a low a/d-conversion rate. An 8-bit a/d-converter yields only a 40 dB dynamic range. In a typical condition monitoring application, this would be too low to reach a satisfactory amplitude resolution. However, the symptoms used as inputs to the classifier have already been pre-processed and made commensurable. Also, when the descriptors and symptoms have been correctly selected and processed, a minor change in machine condition should result in a major change in a symptom value. In fact, we would really not be interested in the least significant decimals of the symptom values.

Most users of the condition monitoring systems will not be interested in the classifier itself, but a simple interface to show the predicted class and its label. In most cases, a simple traffic light type symbol (Figure 55) would be adequate to indicate, if the machine is in a known fault mode (red), unknown anomaly mode (yellow) or known normal mode (green). The light can, for instance, be used in a plant hierarchy display to indicate the global condition of a plant, section or machine.

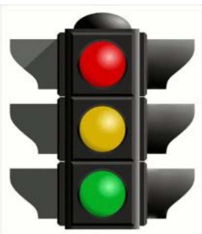

Figure 55: Simple traffic light user interface to display the result of a current state of condition

For those users, who are really interested in the organization of a classifier, an informative led display can be useful (Figure 56). The display with many colour and blinking options can be connected to the output of predictor on a microchip to show immediately the current machine condition while standing next to it.

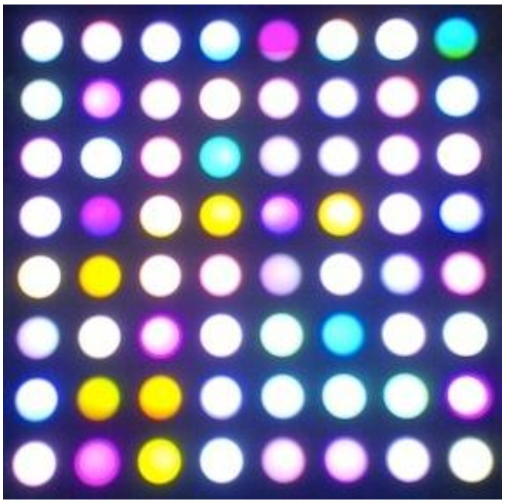

Figure 56: Led display can be used to show the class labels and the prediction of a latest data sample on the classifier map

A more detailed user interface should be available for experts and analysts to allow easy access to the classifier information (Figure 57). The interface should allow easy training and re-training functions to be started on user's initiative. For diagnostics of any novelties, the system should provide access to the symptoms and descriptors.

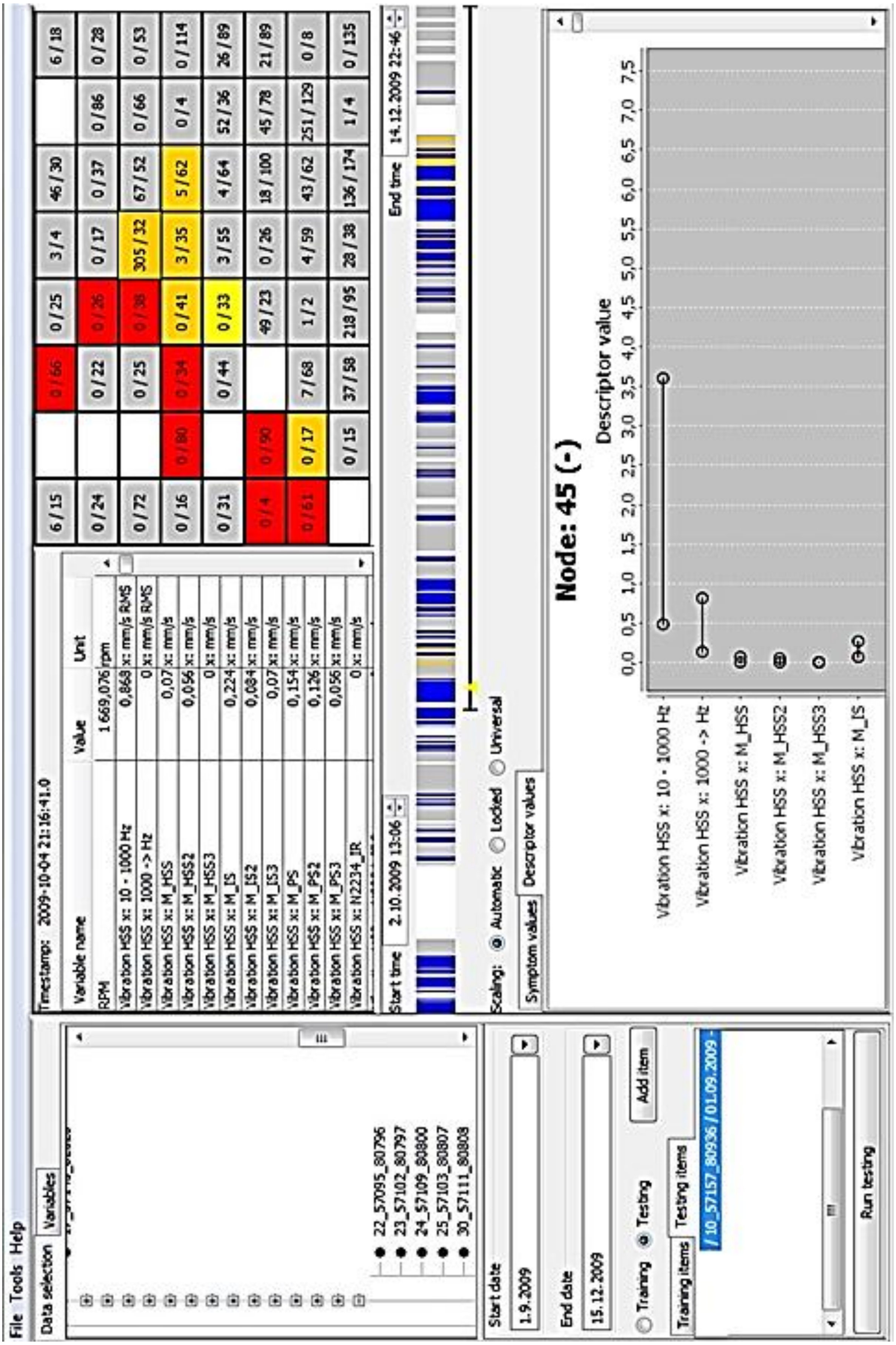

Figure 57: Detailed user interface for an advanced analyst.

9. CONCLUSIONS

The remote diagnostics of machine condition is becoming more popular, because many machines are operated in remote and isolated locations. Therefore, it is necessary to either bring the data to a monitoring centre or perform the diagnostics locally. In cases, where there are numerous items to be monitored and the measurement interval is short, a lot of data is being generated and needs be interpreted. This is often excessive for a human analyst to handle. Automating the diagnostics of rotating machines involves many challenges to be faced.

The thesis describes and demonstrates the prediction of unknown data samples by using a classifier that has been trained with known data samples. Before any prediction, the data shall be pre-processed to improve its quality and representativeness. Poor data quality will distort the prediction and lead to unexpected results. One of the main objectives of data pre-processing is to make data collected from various items at different distant locations comparable. This enables us to utilize data representing fault modes and their severities to benefit the diagnostics and prognostics of other substantially similar items. As a result, a single item does not have to encounter all fault modes in order for them to be interpreted promptly and confidently.

This thesis explains the general principles of classification that are used in the prediction of machine condition. The prediction basically returns in three different outputs: a normal mode, fault mode or anomaly. In practice, there are several differing normal and fault modes. The interpretation of anomalies is often uncertain, but usually the pure detection of an anomaly is valuable. A human analyst will know where to concentrate his efforts. Later, when the anomaly is diagnosed as a normal or fault mode, this interpretation will benefit not only the prediction of the particular item, where it was first detected, but also the prediction of similar items.

The process of continuous learning is effective not only in detection of anomalies and fault modes, but allowing more accurate and confident predictions on the probability of failure and the remaining operational time of the item. The main objective of condition monitoring is to predict the timing of a required maintenance action just before the breakdown. It is very difficult to reach this prediction without previous experiences from similar failure patterns.

The presented experiments clearly provide evidence that a classifier is a reliable tool in detection of anomalies and prediction of fault modes. The results can probably be further improved by enhancing the data pre-processing and by adding more training data from various fault modes.

The confidence level of prediction was tested with data from wind turbine gearboxes. It would be tempting to find out, if the predictions can be further generalized to involve other industrial gearboxes. In general, more research is needed to find out, which monitored items can be combined in the same group, where the same classification rules apply. Most likely, not only the machine type is important but the type of operation is an influential factor in determination, if the machines can be generalized in the same group of items.

The process state is known to have a significant impact on the vibration response of an item. General procedures to evaluate this cause-effect correlation were introduced, but more data and experiments are necessary to verify the relationships between control states and vibration states. The objective of knowing these relationships is to predict the expected vibration response, when the control state is known. This will improve and simplify the prediction.

This thesis introduces the theory and practice for the design of the software engine in order to implement an intelligent diagnostics system for the prediction of machine condition. The individual needs and details for user interface should be separately defined. The interface should be accustomed to serve different types of users.

10. REFERENCES

[Arduino, 2012] Arduino, 2012. http://www.arduino.cc/ [referred 22.10.2012]

[Berthold, 2003] Berthold, M. & Hand, J. H. (Ed.) (2003). Intelligent Data Analysis. Berlin, Germany: Springer-Verlag.

[Berthold, 2010] Berthold, M., Borgelt, C., Höppner, F. & Klawonn, F. (2010). Guide to Intelligent Data Analysis. Berlin, Germany: Springer-Verlag.

[Chisholm, 1966] Chisholm, R. (1966). Theory of Knowledge. New Jersey, USA: Prentice-Hall Inc.

[Churchland, 1966] Churchland, P.S & Sejnowski, T. J. (1996). The Computational Brain. London, England: The MIT Press.

[Digilent, 2012] Digilent Inc, 2012. http://www.digilentinc.com/ [referred 22.10.2012]

[Elfström, 2011] Elfström, J. & Kantola, P. (2011). Condition Monitoring of Transmission in a wind turbine. Promaint Magazine Volume 4. Helsinki, Finland.

[Hagmark, 2011] Hagmark, P-E. (2011). On Three Competing Maintenance actions and the Related Condition Control. International Journal of Quality, Statistics and Reliability. Hindawi Publishing Corporation.

[Han, 2001] Han, J. & Kamber, M. (2001). Data Mining: Concepts and Techniques. London, United Kingdom: Academic Press.

[Hawkins, 2004] Hawkins, J. (2004). On Intelligence. New York, USA: St, Martin's Griffin.

[He, 2009] He, Q., Yan, R., Kong, F. & Du. R. (2009). Machine Condition Monitoring Using Principal Component Representations. Mechacinal Systems and Signal Processing 23 (2009). USA: Elsevier.

[Hintikka, 2005] Hintikka, J. (2005). Knowledge and Belief. London, United Kingdom: King's College Publications.

[Hulkkonen, 2008] Hulkkonen, J. (2008). Minimum Description Length Principle in Novelty Detection. Master's thesis. Helsinki University of Technology.

[Isermann, 2006] Isermann, R. (2006). Fault-Diagnosis Systems. Germany, Berlin: Springer-Verlag.

[ISO 13306, 2010] ISO-EN 13306 (2010). Maintenance. Maintenance terminology.

[ISO 13372, 2004] ISO 13372 (2004). Condition Monitoring and Diagnostics of Machines – Vocabulary.

[ISO 13379, 2003] ISO 13379 (2003). Condition Monitoring and Diagnostics of Machines - General Guidelines on Data Interpretation and Diagnostics Techniques.

[ISO 13381-1, 2004] ISO 13381-1 (2004). Condition Monitoring and Diagnostics of Machines - Prognostics - Part 1: General Guidelines.

[ISO 17359, 2011] ISO 17359 (2011). Condition Monitoring and Diagnostics of Machines - General Guidelines.

[ISO 18436-1, 2004] ISO 18436-1 (2003). Condition Monitoring and Diagnostics of Machines - Requirements for Training and Certification of Personnel - Part 1: Requirements for Certifying Bodies and the Certification Process.

[ISO 18436-2, 2003] ISO 18436-2 (2003). Condition Monitoring and Diagnostics of Machines - Requirements for Training and Certification of Personnel - Part 2: Vibration Condition Monitoring and Diagnostics

[ISO 1940, 2003] ISO 1940 (2003). Mechanical Vibration - Balance Quality Requirements for Rotors in a Constant (Rigid) State - Part 1: Specification and Verification of Balance Tolerances.

[Kohonen, 2001] Kohonen, T. (2001). Self-Organizing Maps. Berlin, Germany:Springer-Verlag.

[Lee, 2011] Lee, J., Ghaffari, M. & Elmeligy, S. (2011). Self-Maintenance and Engineering Immune Systems: Towards smarter machines and manufacturing systems. Annual Reviews in Control. Maryland Heights, MO: Elsevier.

[Lensu, 2002] Lensu, A. (2002). Computationally Intelligent Methods for Qualitative Data Analysis. Jyväskylä, Finland: University of Jyväskylä

[Lumme, 2000] Lumme, V. (2000). Optimizing the Data Processing in Condition Monitoring with Neural Networks. Maintenance Magazine Volume 2. Helsinki, Finland.

[Lumme, 2003] Lumme, V. (2003). Distributed Knowledge in Automated Remote Monitoring. Paper presented at the 17th International Congress & Exhibition on Condition Monitoring and Diagnostic Engineering Management (COMADEM 2003 International). Växjö, Sweden.

[Lumme, 2004] Lumme, V. (2004). Distributed Knowledge in Condition Monitoring. Paper presented at the 8th Euromaintenance Conference. Barcelona, Spain.

[Lumme, 2005] Lumme, V. (2005). Process state classifier. Paper presented at the Automation 05 seminar, Helsinki, Finland.

[Lumme, 2011] Lumme, V. (2011). Diagnosis of Multi-Descriptor Condition Monitoring Data. Paper presented at the International Conference on Prognostics and Heath Management, Denver, CO.

[Lumme, 2012a] Lumme, V. (2012). The principles of classification. In Kadry, S. (ed), Diagnostics and Prognostics of Engineering Systems: Methods and Techniques. IGI Global.

[Lumme, 2012b] Lumme, V. (2012). Evaluating the Confidence Level of Prognostic Predictions. Paper presented at the International Conference on Prognostics and Heath Management, Denver, CO.

[Lumme, FI102857] Lumme, V. & Koikkalainen, P. (1996), Self Learning Method in the Condition Monitoring and Diagnostics of Rotating Machines, FI102857, Helsinki, Finland: Finnish Patent Office

[Lumme, FI115486] Lumme, V. (2005), Method in Handing of Samples, FI115486, Helsinki, Finland: Finnish Patent Office

[Lumme, FI118746] Lumme, V. (2005), Method in Monitoring of an Industrial Process. FI118746, Helsinki, Finland: Finnish Patent Office

[Lumme, EP1292812] Lumme, V. & Seppä, J. (2001), Method in Monitoring the Condition of Machines, EP1292812, Munich, Germany: European Patent Office.

[Miettinen, 1997] Miettinen, J., Lumme, V. & Siekkinen, V (1997) Detection of Fault Modes of Gear Drive Using Neural Net, Paper presented at the 10th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management. Espoo, Finland

[Mikkonen, 2009] Mikkonen, H., Jantunen, E., Miettinen, J., Leinonen, P., Kautto, J. & Lumme, V. (2009). Diagnostics. Diagnostiikka. In Mikkonen, H. (ed), Condition Based Maintenance. Kuntoon perustuva kunnossapito (pp. 281-368). Helsinki, Finland: KP-Media Oy.

[Milne, 1996] Milne, R. (1996). Knowledge Based Systems for Maintenance management. In Rao, B. K. N. (ed), Handbook of Condition Monitoring (pp. 377-393). Oxford, United Kingdom: Elsevier Advanced Technology.

[Nohynek, 2004] Nohynek, P. & Lumme V. (2004). Kunnonvalvonnan värähtelymittaukset. Rajamäki, Finland: KP-media Oy

[PSK 5970, 2004] PSK 5970 (2004). XML Data Transmission. Description of a Case. Helsinki, Finland: PSK Standardization

[PSK 8301, 2007] PSK 8301 (2007). Shaft Alignment. Helsinki, Finland: PSK Standardization

[Pöyhönen, 2004] Pöyhönen, S. (2004). Support Vector Machine Based Classification in Condition Monitoring of Induction Motors. Helsinki University of Technology.

[Setälä, 2004] Setälä, T., Miettinen, M & Lumme V (2004). Correlation between Process Variables & Vibration Data using Neural Network. Paper presented at the 17th International Congress & Exhibition on Condition Monitoring and Diagnostic Engineering Management (COMADEM 2004 International), Cambridge, UK.

[Sheng, 2011] Sheng, C., Li, Z., Qin, L., Guo, Z. & Zhang Y. (2011). Recent Progress on Mechanical Condition Monitoring and Fault Diagnosis. Procedia Engineering 15. USA: Elsevier.

[Sorsa, 1991] Sorsa, T., Koivo, H. & Koivisto, H. (1991). Neural Networks in Process Fault Diagnosis. IEEE Transactions on Systems, Man, and Cybernetics 21. USA, New York: IEEE.

[Wong, 2006] Wong, M., Jack, L. & Nandi, A. (2006). Modified self-organising map for automated novelty detection applied to vibration signal monitoring. Mechanical Systems and Signal Processing 20. USA, New York: Elsevier.

[Yang, 2004] Yang, B., Han, T. & An, J. (2004). ART-KOHONEN neural network for fault diagnosis of rotating machinery. Mechanical Systems and Signal Processing 18. USA, New York: Elsevier.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-2963-4
ISSN 1459-2045